AD-A052 733 CHARLES STARK DRAPER LAB INC CAMBRIDGE MA F/G 9/2
JOVIAL STRUCTURED DESIGN DIAGRAMMER (JSDD). VOLUME III. PROGRAM--ETC(U)
FEB 78 G GODDARD, M WHITWORTH, E STROVINK F30602-76-C-0408
UNCLASSIFIED R-1120-VOL-3-PT-3 RADC-TR-78-9-VOL-3-PT-3 NL

1 OF 3
AD
A052733

2

RADC-TR-78-9, Vol III, Part 3 (of four)
Final Technical Report
February 1978

JOVIAL STRUCTURED DESIGN DIAGRAMMER (JSDD), Volume III
Program Description, Part 3

G. Goddard
M. Whitworth
E. Strovink

The Charles Stark Draper Laboratory, Inc.

Approved for public release; distribution unlimited.

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, New York 13441

Because of the size of this volume, it has been divided into four parts. Part 1 contains pages 1/2 - 123, 649 - 657, Part 2 contains pages 124 - 344. Part 3 contains pages 345 - 592, Part 4 contains pages 593 - 648.

This report has been reviewed by the RADC Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RADC-TR-78-9, Vol III, Part 3 has been reviewed and is approved for publication.

APPROVED: Donald VanAlstine

DONALD VANALSTINE
Project Engineer

APPROVED: Wendall C Bauman

WENDALL C. BAUMAN, Colonel, USAF
Chief, Information Sciences Division

FOR THE COMMANDER: John P. Huss

JOHN P. HUSS
Acting Chief, Plans Office

If your address has changed or if you wish to be removed from the RADC mailing list, or if the addressee is no longer employed by your organization, please notify RADC (ISIM) Griffiss AFB NY 13441. This will assist us in maintaining a current mailing list.

Do not return this copy. Retain or destroy.

(18) RADC (19) TR-78-9-VOL-3-PT-3

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER RADC-TR-78-9, Vol III, Part 3 (of four) / 2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE ~~(and Subtitle)~~ (6) JOVIAL STRUCTURED DESIGN DIAGRAMMER (JSDD). ~~Program Direction~~ Volume III. Program Description. Part 3.	(9) 5. TYPE OF REPORT & PERIOD COVERED Final Technical Report, Sep[redacted] 76 - Oct[redacted] 77,
	6. PERFORMING ORG. REPORT NUMBER R-1120-VOL-3-PT-3
7. AUTHOR(s) (10) G. Goddard, M. Whitworth E. Strovink	(14) 8. CONTRACT OR GRANT NUMBER(s) (15) F30602-76-C-0408
9. PERFORMING ORGANIZATION NAME AND ADDRESS The Charles Stark Draper Laboratory, Inc. 555 Technology Square Cambridge MA 02139	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS P.E. 62702C J.O. 55811412
11. CONTROLLING OFFICE NAME AND ADDRESS Rome Air Development Center (ISIM) Griffiss AFB NY 13441	(11) 12. REPORT DATE Feb[redacted]978
	13. NUMBER OF PAGES 248 (12) 252P.
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Same (16) 5581 (17) 14	15. SECURITY CLASS. (of this report) UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Same

18. SUPPLEMENTARY NOTES

RADC Project Engineer: Donald VanAlstine (ISIM)

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Structured programming	Preprocessor
Structured design diagram	Flowcharter
Structured extension	JOVIAL J3
Parse	Invocation diagram
Parser generator	

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The report presents a detailed description of the JOVIAL Structured Design Diagrammer program implementation for purposes of maintaining or modifying the system.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

408 386 JOB

Section 8

Phase 2 JOVIAC Structured Design and Invocation Diagram~~s~~er.

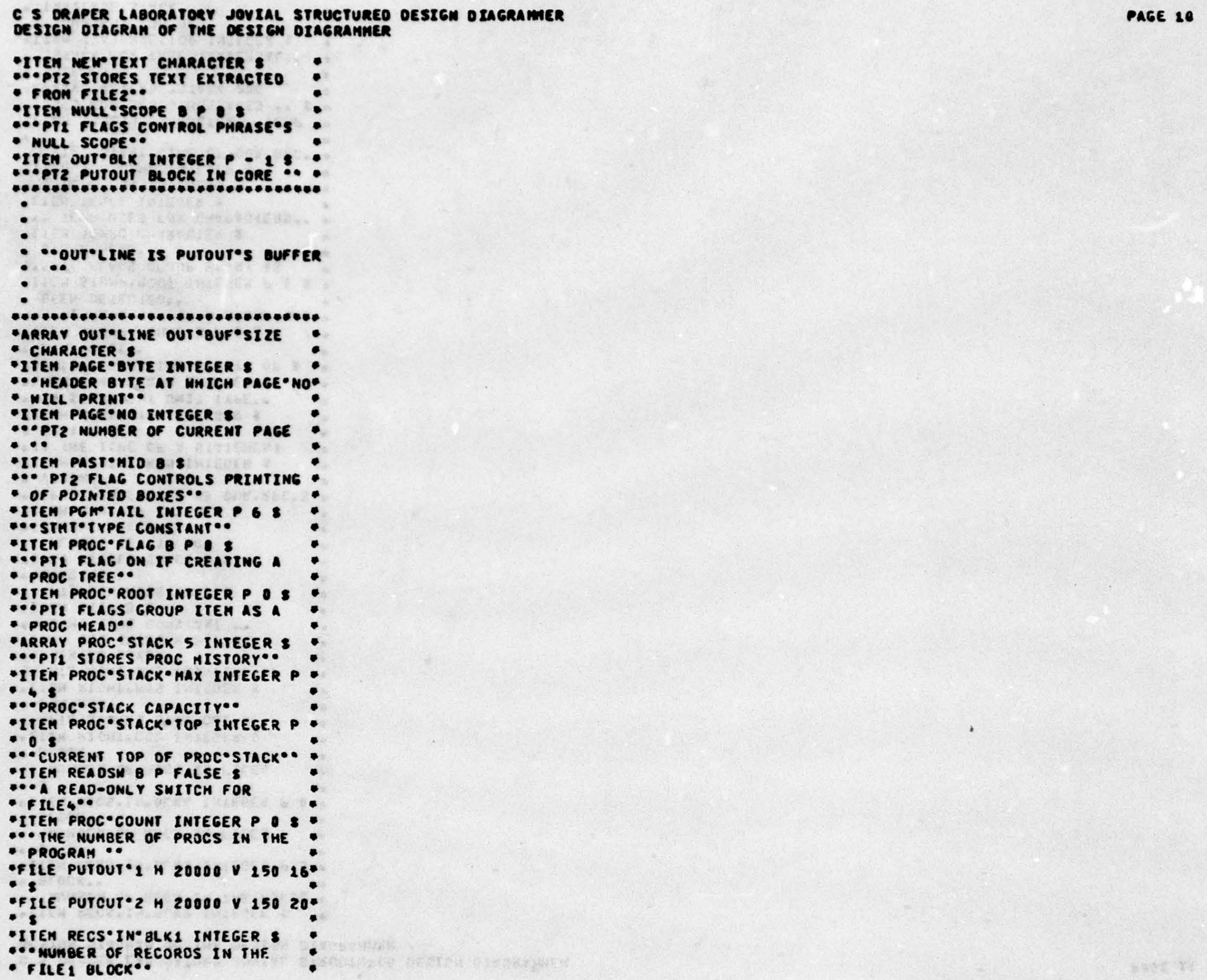
```
C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER                                          PAGE 1


THIS LISTING CONSISTS OF OUTPUT FROM
THE CHARLES STARK DRAPER LABORATORY'S JOVIAL J3
STRUCTURED DESIGN DIAGRAMMER.


PRINCIPAL DESIGNERS AND IMPLEMENTORS

GARY W. GOODARD, CSDL STAFF
MARK H. WHITWORTH, CSDL STAFF
ERIC F. STROVINK, GRADUATE STUDENT, M.I.T.
COMPUTER SCIENCE DIVISION
THE CHARLES STARK DRAPER LABORATORY, INC., CAMBRIDGE, MA.
```

TABLE OF CONTENTS

TABLE OF CONTENTS

```
***********
*START $  *
***********
 *
 *
********************************
*DEFINE CHARACTER ''H 150'' $  *
*DEFINE F1'BLKSIZ ''78'' $     *
*DEFINE F2'BLKSIZ ''12'' $     *
*DEFINE F3'BLKSIZ ''22'' $     *
*DEFINE F4'BLKSIZ ''105'' $    *
*DEFINE FALSE ''0'' $          *
*DEFINE INTEGER ''I 36 S'' $   *
*DEFINE OUT'BUF'SIZE ''1000'' $*
*                              *
*DEFINE TRUE ''1'' $           *
*ITEM BOTTOM'LINE INTEGER $    *
*''PI2. STORES STOP'LINE OF    *
* CUR'REC REC''                *
*ITEM BLOCK'DELIM INTEGER P 5 $*
*                              *
*'' A STMT'TYPE CONSTANT ''    *
*ITEM BOX'TAIL INTEGER P 2 $   *
*''LENGTH OF CONTROL'3 TAIL '' *
*ITEM COMMENT'1 INTEGER P 2 $  *
*'' A STMT'TYPE CONSTANT''     *
*ITEM COMMENT'2 INTEGER P 3 $  *
*'' A STMT'TYPE CONSTANT''     *
********************************
 *
 *
 *
 * ''CONTROL'1 THRU CONTROL'4 ARE
 *  STMT'TYPE CONSTANTS FOR ''
 * ''CONTROL PHRASES.''
 *
 *
********************************
*ITEM CONTROL'1 INTEGER P 8 $  *
*''PROGRAM, PROC OR CLOSE      *
* HEAD''                       *
*ITEM CONTROL'2 INTEGER P 9 $  *
*''IF, FOR OR DO CLAUSE''      *
*ITEM CONTROL'3 INTEGER P 10 $ *
*''IFEITH CLAUSE OR CASE HEAD''*
*                              *
*ITEM CONTROL'4 INTEGER P 11 $ *
*''ORIF CLAUSE OR INSTANCE ''  *
*ITEM CUR'GROUP INTEGER P 0 $  *
*''INDEX INTO GROUP OF CURRENT *
* TREE''                       *
*ITEM CUR'REC INTEGER P 0 $    *
*''CURRENT FILE 3 RECORD ''    *
*ITEM DELIM'COMMENT INTEGER P 4*
* $                            *
*'' STMT'TYPE CONSTANT ''      *
```

```
*ITEM DISP INTEGER $            *
***PT2 DISPLACEMENT FOR         *
* RELATIVE LINE NUMBERING**     *
*ITEM DISPLAY*LINES INTEGER P 3*
* $                             *
***LINES SPANNED BY A STUMP REF*
* BOX**                         *
*ITEM DISPLAY*WIDTH INTEGER P 6*
* $                             *
***WIDTH OF STUMP REF BOX**     *
*ITEM END*SCOPE INTEGER P 1 $   *
***STMT*TYPE CONSTANT **        *
*ITEM EOFILE B P 0 $            *
***FILE 1 END OF FILE FLAG**    *
*ITEM EXTRA*BLOCK B P FALSE $   *
***FLAGS EXTRA PUTOUT FILE      *
* BLOCK**                       *
*********************************
 *
 *
 *
 * ** F1*BUF IS THE FILE 1 BUFFER
 *  **
 *
 *
*********************************
*TABLE F1*BUF R F1*BLKSIZ S N $*
*                               *
*BEGIN                          *
*ITEM FIELD1 INTEGER $          *
*ITEM FIELD2 INTEGER $          *
*ITEM FIELD3 INTEGER $          *
*ITEM FIELD4 INTEGER $          *
*END                            *
*********************************
 *
 *
 *
 * ** F2*BUF IS THE FILE 2
 *   BUFFER**
 *
************ ********************
*TABLE F2*BUF R F2*BLKSIZ S N $*
*                               *
*BEGIN                          *
*ITEM F2*LINE CHARACTER $       *
*END                            *
*ITEM F3*AVAIL INTEGER P 1 $    *
***THE NEXT EMPTY FILE 3 RECORD*
* **                            *
*********************************
 *
 *
 *
 * **F3*BUF IS THE FILE 3 BUFFER
 *  **
 *
 *
**********************************
*TABLE F3*BUF R F3*BLKSIZ S N $*
*                                *
*BEGIN                           *
*ITEM F4*BEGIN INTEGER $         *
*ITEM F4*END INTEGER $           *
*ITEM STMT*UNIT INTEGER $        *
*ITEM START*COL INTEGER $        *
*ITEM BLOCK*WIDTH INTEGER $      *
*ITEM MIDPT INTEGER $            *
*ITEM STOP*COL INTEGER $         *
*ITEM START*LINE INTEGER $       *
*ITEM LINES INTEGER $            *
*ITEM STOP*LINE INTEGER $        *
*ITEM H*PTR INTEGER $            *
*ITEM V*PTR INTEGER $            *
*ITEM BACK*H INTEGER $           *
*ITEM BACK*V INTEGER $           *
*END                             *
**********************************
 *
 *
 *
 * **F4*BUF  IS THE FILE 4 BUFFER
 *  **
 *
 *
**********************************
*TABLE F4*BUF R F4*BLKSIZ S N $*
* TABLE F4*BUF R F4*BLKSIZ S N   *
*BEGIN                           *
*ITEM F2*PTR INTEGER $           *
*ITEM LINES*OUT INTEGER $        *
*ITEM MAX*LINE*LNGTH INTEGER $   *
*ITEM F2*REC INTEGER $           *
*ITEM F2*BYTE INTEGER $          *
**********************************
 *
 *
 *
 * ** OVERLAY THE DISJOINT
 *  FIELDS**
 *
 *
*********************************
*OVERLAY LINES*OUT = F2*REC $   *
*OVERLAY MAX*LINE*LNGTH =       *
* F2*BYTE $                     *
*END                            *
*********************************
 *
 *
 *
 * ** GROUP IS THE FILE 3 TREE
 *   POINTER **
 *
 *
***********************************
*TABLE GROUP V 500 P N $          *
*BEGIN                            *
*ITEM F3'REC INTEGER $            *
*ITEM FROM'PAGE INTEGER $         *
*ITEM NEXT INTEGER $              *
*ITEM PAGE'REF INTEGER $          *
*ITEM PROC'NAME CHARACTER $       *
*END                              *
*ITEM F2'RECS INTEGER $           *
*''NO. OF FILE 2 RECS SPANNED     *
* BY A STMT UNIT''                *
*ITEM F1'BLK INTEGER P - 1 $      *
*''NUMBER OF FILE 1 BLOCK IN      *
* CORE ''                         *
*ITEM F2'BLK INTEGER P - 1 $      *
*''NUMBER OF FILE 2 BLOCK IN      *
* CORE ''                         *
*ITEM F3'BLK INTEGER P - 1 $      *
*''NUMBER OF FILE 3 BLOCK  IN     *
* CORE''                          *
*ITEM F4'BLK INTEGER P - 1 $      *
*''NUMBER O F FILE 4 BLOCK IN     *
* CORE ''                         *
*ITEM F3'EMPTY INTEGER P 0 $      *
*''NEXT EMPTY FILE 3 BLOCK ''     *
*ITEM F4'EMPTY INTEGER P 0 $      *
*''NEXT EMPTY FILE 4 BLOCK ''     *
*ITEM F4'START INTEGER $          *
*''PT2 STORES F4'BEGIN OF         *
* CUR'REC ''                      *
*ITEM F4'STOP INTEGER $           *
*''PT2 STORES F4'END OF CUR'REC   *
* ''                              *
*FILE FILE1 B 10000 R 313 V(A)    *
* V(B) V(C) V(D) V(EOF1) 11 $     *
*FILE FILE2 H 10000 V 1806 13 $   *
*                                 *
*FILE FILE3'1 B 10000 V 309 14    *
* $                               *
*FILE FILE3'2 B 10000 V 309 18    *
* $                               *
*FILE FILE4'1 B 10000 R 316 15    *
* $                               *
*FILE FILE4'2 B 10000 R 316 19    *
* $                               *
*ITEM FILE2'INDEX INTEGER $       *
*''PT1 INDEX OF STMT UNIT IN      *
* FILE 2 ''                       *
***********************************
 *
 *
 *
 *
 *  ''FILE3'INCLUSION IS A FLAG
 *   INDICATING WHETHER A FILE 3
 *  REC SHOULD ••
 * •• BE CREATED FOR THE CURRENT
 *  FILE 1 REC (PT1)••
 *
 *
**********************************
*ITEM FILE3•INCLUSIJN B $        *
*FILE FINAL•OUT H 20000 V 150    *
* 17 $                           *
*ITEM FIRST•4 INTEGER $          *
*•• PT1  INDEX OF 1ST FILE 4     *
* REC OF SET ••                  *
*ITEM FIRST•INVOC B P TRUE $     *
*••FLAGS FIRST CALL TO           *
* EXTRACT•TEXT••                 *
*ITEM GROUF•AVAIL INTEGER P 1 $*
*                                *
*••INDEX OF NEXT EMPTY SPACE IN*
* GROUP ••                       *
*ITEM GROUP•MAX INTEGER P 499 $*
*                                *
*•• THE SIZE OF GROUP ••         *
*ARRAY GROUP•STACK 100 INTEGER *
* $                              *
*••PT2 FOR STORAGE OF TREE       *
* HISTORY••                      *
*ITEM HEADROOM INTEGER P 4 $     *
*••LINES SPANNED BY PAGE         *
* HEADING••                      *
*ITEM H•FATHER INTEGER $         *
*••PT2 STORES BACK•V OF CUR•REC*
* ••                             *
*ITEM H•SON INTEGER $            *
*••PT2 STORES H•PTR OF CUR•REC *
* ••                             *
*ITEM H•SPACE•1 INTEGER P 4 $   *
*•• HORIZONTAL SPACING CONSTANT*
* ••                             *
*ITEM H•SPACE•2 INTEGER P 2 $   *
*•• HORIZONTAL SPACEING          *
* CONSTANT ••                    *
*ITEM LAST•4 INTEGER P - 1 $     *
*••PT1 INDEX OF LAST FILE4 REC *
* OF A SET•• ••PT1 INDEX OF LAS*
*ITEM LAST•F1 INTEGER P - 1 $   *
*••PT1 INDEX OF LAST FILE1 REC *
* READ••                         *
**********************************
 *
 *
 *
 * ••LAST•LINE IS USED IN PART 1
 *  TO STORE THE LAST RELATIVE ••
 * •• LINE NUMBER IN THE CURRENT
 *  TREE. IN PART2 IT IS THE
 *  ABSOLUTE ••
 * •• LAST LINE WHICH HAS BEEN
 *  OUTPUT ''
 *
 *
**********************************
*ITEM LAST'LINE INTEGER $        *
*ITEM LAST'PROC INTEGER P - 1 $*
*                                *
*''INDEX IN GROUP OF THE LAST    *
* PROC ''                        *
*ITEM LAST'STUMP INTEGER P 0 $   *
*'' INDEX IN GROUP OF THE LAST   *
* STUMP ''                       *
*ARRAY LAYOUT'STACK 100 INTEGER*
* $                              *
*'' STORES HISORY OF             *
* LAST'LINE''                    *
*ITEM LAYOUT'STACK'MAX INTEGER   *
* P 99 $                         *
*''CAPACITY OF LAYOUT'STACK''    *
*ITEM LAYOUT'STACK'TOP INTEGER   *
* P 0 $                          *
*'' CURRENT TOP OF               *
* LAYOUT'STACK''                 *
*ITEM LEFT'COL INTEGER $         *
*''PT2 START'COL OF CUR'REC''    *
*ITEM LINE'NO INTEGER $          *
*''PT2 THE NUMBER OF THE LINE    *
* BEING OUTPUT''                 *
*ITEM MAX'FOUTPUT INTEGER P - 1*
* $                              *
*''INDEX OF LAST PUTOUT BLOCK    *
* ON DISK''                      *
*ITEM MESSAGE CHARACTER $        *
*'' STORES ERROR AND DEBUG       *
* MESSAGES''                     *
*ITEM MIDPOINT INTEGER $         *
*''STORES MIDPT OF CUR'REC''     *
**********************************
 *
 *
 *
 *  ''NAME'BYTE IS THE BYTE OF
 *  HEADER AT WHICH THE MODULE
 *  NAME PRINTS''
 *
 *
**********************************
*ITEM NAME'BYTE INTEGER $        *
*ITEM NEW'FILE3 B P 0 $          *
*''FLAGS FILE3'1 OR FILE3'2 AS   *
* MOST RECENT''                  *
*ITEM NEW'FILE4 B P 0 $          *
*''FLAGS FILE4'1 OR FILE4'2 AS   *
* MOST RECENT''                  *
*ITEM NEW'OUT B P 0 $            *
*''FLAGS PUTOUT'1 OR PUTOUT'2    *
* AS MOST RECENT''               *
```



```
*ITEM NEW'TEXT CHARACTER $     *
*''PT2 STORES TEXT EXTRACTED   *
* FROM FILE2''                 *
*ITEM NULL'SCOPE B P 0 $       *
*''PT1 FLAGS CONTROL PHRASE'S  *
* NULL SCOPE''                 *
*ITEM OUT'BLK INTEGER P - 1 $  *
*''PT2 PUTOUT BLOCK IN CORE '' *
********************************
 *
 *
 *
 * ''OUT'LINE IS PUTOUT'S BUFFER
 *  ''
 *
 *
********************************
*ARRAY OUT'LINE OUT'BUF'SIZE   *
* CHARACTER $                  *
*ITEM PAGE'BYTE INTEGER $      *
*''HEADER BYTE AT WHICH PAGE'NO*
* WILL PRINT''                 *
*ITEM PAGE'NO INTEGER $        *
*''PT2 NUMBER OF CURRENT PAGE  *
* ''                           *
*ITEM PAST'MID B $             *
*'' PT2 FLAG CONTROLS PRINTING *
* OF POINTED BOXES''           *
*ITEM PGM'TAIL INTEGER P 6 $   *
*''STMT'TYPE CONSTANT''        *
*ITEM PROC'FLAG B P 0 $        *
*''PT1 FLAG ON IF CREATING A   *
* PROC TREE''                  *
*ITEM PROC'ROOT INTEGER P 0 $  *
*''PT1 FLAGS GROUP ITEM AS A   *
* PROC HEAD''                  *
*ARRAY PROC'STACK 5 INTEGER $  *
*''PT1 STORES PROC HISTORY''   *
*ITEM PROC'STACK'MAX INTEGER P *
* 4 $                          *
*''PROC'STACK CAPACITY''       *
*ITEM PROC'STACK'TOP INTEGER P *
* 0 $                          *
*''CURRENT TOP OF PROC'STACK'' *
*ITEM READSW B P FALSE $       *
*''A READ-ONLY SWITCH FOR      *
* FILE4''                      *
*ITEM PROC'COUNT INTEGER P 0 $ *
*''THE NUMBER OF PROCS IN THE  *
* PROGRAM ''                   *
*FILE PUTOUT'1 H 20000 V 150 16*
* $                            *
*FILE PUTCUT'2 H 20000 V 150 20*
* $                            *
*ITEM RECS'IN'BLK1 INTEGER $   *
*''NUMBER OF RECORDS IN THF    *
* FILE1 BLOCK''                *
```

```
•ITEM RECS•IN•BLK2 INTEGER $   •
•••NUMBER OF RECS IN THE FILE2 •
• BLOCK••                      •
•ITEM RECS•IN•BLK3 INTEGER P 0 •
• $                            •
•••NUMBER OF RECS IN FILE3     •
• BLOCK••                      •
•ITEM RECS•IN•BLK4 INTEGER P 0 •
• $                            •
•••NUMBER OF RECS IN FILE4     •
• BLOCK••                      •
•ITEM RIGHT•COL INTEGER $      •
•••PT2 STORES STOP•COL OF      •
• CUR•REC••                    •
•ITEM RIGHT•POS INTEGER $      •
•••PT2 MONITORS PRINTING OF    •
• POINTED BOXES••              •
•ITEM SEQ INTEGER P 7 $        •
•••STMT•TYPE CONSTANT ••       •
•ITEM SKIP B $                 •
•••PT2 CONTROLS DOUBLE         •
• SPACING••                    •
•ITEM SINGLE•SPACE B $         •
•••ALWAYS SET TO NOT           •
• DOUBLE•SPACE••               •
•ITEM SON•TOP INTEGER $        •
•••PT2 START•LINE OF CUR•REC•S •
• V•SON••                      •
•ITEM STMT•TOKEN INTEGER $     •
••• THE TYPE OF A STATEMENT    •
• UNIT••                       •
•ITEM STMT•TYPE INTEGER $      •
•••A STATEMENT UNIT TYPE••     •
•ITEM STMT•LNGTH INTEGER $     •
•••PT1 STORES THE LENGTH OF A  •
• STMT UNIT••                  •
•ITEM STUMP•FOUND B P 0 $      •
•••PT1 FLAG SET IF A STUMP HAS •
• BEEN DETECTED••              •
•ITEM STUMP•ROOT INTEGER P 1 $ •
•••PT1 FLAGS GROUP ENTRY AS    •
• STUMP HEAD••                 •
•ITEM TEMPC CHARACTER $        •
••• TEMP USED FOR CHARACTERS•• •
•ITEM TEMPI INTEGER $          •
••• TEMP FOR INTEGERS••        •
•ITEM TOP•LINE INTEGER $       •
•••PT2  START•LINE OF CUR•REC•••
•                              •
•ITEM T•MESS CHARACTER ••TEMP  •
• USED FOR DEBUG MESSAGES •• $ •
•ARRAY TRAVERSE•STACK 200      •
• INTEGER $                    •
•••STACK FOR TREE TRAVERSAL••  •
•ITEM TRAVERSE•TOP INTEGER $   •
••• CURRENT TOP OF             •
• TRAVERSE•STACK••             •
```

```
G S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER                                                         PAGE 12
DESIGN DIAGRAM OF THE DESIGN DIAGRAMMER

*ITEM V'FATHER INTEGER $       *
*'' PT2 BACK'V OF CUR'REC''    *
*ITEM V'SON INTEGER $          *
*''PTZ V'PTR OF CUR'REC''      *
*ITEM V'SPACE INTEGER P 3 $    *
*''VERTICAL SPACING CONSTANT'' *
*ITEM WIDTH INTEGER $          *
*'' STORES WIDTH OF CURRENT    *
* TEXT LINE OR BOX''           *
*ITEM WRITE3 B $               *
*'' FILE 3'S WRITE SWITCH''    *
*ITEM WRITE4 B $               *
*''FILE 4'S WRITE SWITCH''     *
********************************
 *
 *
 *
 * ** MAIN PHASE 2 ROUTINE **
 *
 *
***************
*INITIALIZE $ *
***************
 *
 *
**************
*IF DEBUG1 $ *     ********************************
*''5''        *---*MESSAGE = 13H(INIT COMPLETE) $*
*             *   *                              *
**************    *OUT(MESSAGE, MESS'SW) $       *
 *                ********************************
 *
********************************
*DO UNTIL (EOFILE OR CUR'REC EQ *    ***************
* 0)                            *---*GET'F1'REC $ *
*                               *    ***************
********************************      *
 *                                    *
 *                                    ******************    **********************
 *                                    *IF NOT EOFILE $ *---*CREATE'FILE4'RECS $ *
 *                                    ******************    **********************
 *                                                           *
 *                                                           *
 *                                                          ***********************    *****************
 *                                                          *IF FILE3'INCLUSION $ *---*UPDATE'FILE3 $ *
 *                                                          ***********************    *****************
 *
 *
**************
*IF DEBUG1 $ *     ****************************
*''5''        *---*MESSAGE = 9H(END PART1) $ *
*             *   *OUT(MESSAGE, MESS'SW) $   *
**************    ****************************
 *
 *
**********
*PART2 $ *
**********
```

```
**************
*IF DEBUG1 $ *    ******************************
*''5''        *---*MESSAGE = 10H(PART2 DONE) $ *
*            *    *OUT(MESSAGE, MESS'SW) $     *
**************    ******************************
 *
 *
**************
*IF DEBUG1 $ *    ********************************
*''9''        *---*MESSAGE = 10H(LAST LINE ) $   *
*            *    *T'MESS = IFORMAT(LAST'LINE) $ *
**************    *MESSAGE = CAT(MESSAGE, T'MESS)*
 *                * $                            *
 *                *OUT(MESSAGE, MESS'SW) $       *
 *                *MESSAGE = 12H(START BYTING) $ *
 *                *OUT(MESSAGE, MESS'SW) $       *
 *                ********************************
 *
************
*BYTE'EM $ *
************
 *
 *
**************************   **************************
*IF PROC'STACK'TOP NQ 0 $ *---*MESSAGE = 16H(PROC STACK*
**************************   * ERROR) $                *
 *                           *PH2ERR(MESSAGE) $        *
 *                           **************************
 *
 *
 * ''THE STRING PACKAGE ''
 *
 *
*********
*TERM $ *
*********
```

```
C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER
DESIGN DIAGRAM OF ACCESS1

*************************
*PROC ACCESS1(REC'NO) $  *
*************************
 .
 .
 .
 .  ''THE FUNCTION ACCESS1
 .   PROVIDES THE INTERFACE FOR
 .   FILE 1. ITS PARAMETER IS AN
 .   ABSOLUTE RECORD NUMBER.
 .   ACCESS1 READS THE APPROPRIATE
 .   BLOCK OF FILE 1 RECORDS INTO
 .   CORE AND RETURNS THE INDEX
 .   INTO FILE 1'S BUFFER OF THE
 .   DESIRED RECORD.''
 .
********************************
*ITEM ACCESS1 INTEGER $        *
*''INDEX INTO F1'BUF OF        *
* RECORD''                     *
*ITEM BLOCK'NO INTEGER $       *
*'' FILE1 BLOCK IN WHICH RECORD*
* RESIDES''                    *
*ITEM REC'NO INTEGER $         *
*''ABSOLUTE NUMBER OF THE      *
* RECORD''                     *
********************************
 .
 .
**************
*IF DEBUG2 $  *     ********************************
*''7''        *---*MESSAGE = 9H(ACCESS 1 ) $     *
*             *   *T'MESS = IFORMAT(REC'NO) $    *
**************    *MESSAGE = CAT(MESSAGE, T'MESS)*
 .                * $                            *
 .                *OUT(MESSAGE, MESS'SW) $       *
 .                ********************************
 .
***************************
*REMQUO(REC'NO, F1'BLKSIZ =*
* BLOCK'NO, ACCESS1) $     *
***************************
 .
 .
**************************   ***************************
*IF F1'BLK NQ BLOCK'NO $ *---*POS(FILE1) = BLOCK'NO $  *
**************************   *INPUT FILE1 RECS'IN'BLK1,*
 .                           * F1'BUF $                *
 .                           *F1'BLK = BLOCK'NO $      *
 .                           ***************************
 .                             .
 .                             .
 .                             .
 .                             .
```

```
C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER                                             PAGE 15
DESIGN DIAGRAM OF ACCESS1

 *                              **************
 *                              *IF DEBUG2 $  *    **********************************
 *                              *"*7"*         *---*MESSAGE = 12H(ACCESS1 GET ) $ *
 *                              *             *    *T"MESS = IFORMAT(BLOCK"NO) $    *
 *                              **************     *MESSAGE = CAT(MESSAGE, T"MESS)*
 *                                                 * $                              *
 *                                                 *OUT(MESSAGE, MESS"SW) $         *
 *                                                 **********************************
 *
********************************
*IF ACCESS1 GQ RECS"IN"BLK1 OR *     *******************
* FILE1 EQ V(EOF1) $             *---*ACCESS1 = - 1 $ *
*                               *    *""SIGNAL EOF "" *
********************************     *******************
 *
 *
***********
*RETURN $ *
***********
```

```
DESIGN DIAGRAM OF ACCESS2

*************************
*PROC ACCESS2(REC'NO) $  *
*************************
 *
 *
 *
 * ''ACCESS2 ACCEPTS AN ABSOLUTE
 *   RECORD NUMBER AND MAKES SURE
 *   THAT THE FILE2 BLOCK
 *   CONTAINING THAT RECORD IS IN
 *   CORE. ACCESS2 RETURNS THE
 *   INDEX INTO F2'BUF OF THE
 *   RECORD. ''
 *
 *
********************************
*ITEM ACCESS2 INTEGER $        *
*''THE INDEX INTO F2'BUF OF THE*
* RECORD''                     *
*ITEM BLOCK'NO INTEGER $       *
*''THE FILE2 BLOCK CONTAINING  *
* THE RECORD''                 *
*ITEM REC'NO INTEGER $         *
*''THE ABSOLUTE NUMBER OF THE  *
* FILE2 RECORD''               *
********************************
 *
 *
*******************    ********************************
*IF DEBUG3 ''7'' $  *---*MESSAGE = 9H(ACCESS 2 ) $     *
*******************    *T'MESS = IFORMAT(REC'NO) $    *
 *                     *MESSAGE = CAT(MESSAGE, T'MESS)*
 *                     * $                            *
 *                     *OUT(MESSAGE, MESS'SW) $       *
 *                     ********************************
 *
***************************
*REMQUO(REC'NO, F2'BLKSIZ =*
* BLOCK'NO, ACCESS2) $    *
***************************
 *
 *
*************************      *************************
*IF F2'BLK NQ BLOCK'NO $  *---*POS(FILE2) = BLOCK'NO $ *
*************************      *INPUT FILE2 RECS'IN'BLK2,*
 *                             * F2'BUF $                *
 *                             *F2'BLK = BLOCK'NO $      *
 *                             **************************
 *                               *
 *                               *
 *                             **************
 *                             *IF DEBUG3 $  *    ********************************
 *                             *''7''        *---*MESSAGE = 12H(ACCESS2 GET ) $ *
 *                             *             *    *T'MESS = IFORMAT(BLOCK'NO) $  *
 *                             **************    *MESSAGE = CAT(MESSAGE, T'MESS)*
 *                                               * $                            *
```

```
C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER                                    PAGE 17
DESIGN DIAGRAM OF ACCESS2

                                                   *OUT(MESSAGE, MESS'SW) $         *
 *                                                 **********************************
 *
********************************   ******************
*IF ACCESS2 GQ RECS'IN'BLK2 $  *---*ACCESS2 = - 1 $ *
********************************   *''SIGNAL EOF '' *
 *                                 ******************
 *
************
*RETURN $ *
************
```

```
DESIGN DIAGRAM OF ACCESS3

*************************
*PROC ACCESS3(REC*NO) $  *
*************************
  *
  *
  *
  * ** THE FUNCTION ACCESS3 IS THE
  *   INTERFACE FOR FILE3 ITS
  *   PARAMETER IS THE ABSOLUTE
  *   RECORD NUMBER OF A FILE 3
  *   RECORD. ACCESS3 DETERMINES IF
  *   THE BLOCK CONTAINING  THE
  *   RECORD IS IN CORE. IF IT IS,
  *   THEN THE INDEX INTO THE
  *   BUFFER OF THE RECORD IS
  *   RETURNED. IF THE BLOCK IS
  *   EXISTS BUT IS NOT IN CORE,
  *   THEN WRITE3 IS EXAMINED. IF
  *   WRITE3 IS ON(EQUAL TO 1)
  *   THEN, THE BUFFER MUST BE
  *   WRITTEN TO DISK (BY TRANSS EX
  *   TRANSFER*WRITE3). ACCESS3
  *   READS THE DESIRED BLOCK INTO
  *   THE BUFFER AND RETURNS THE
  *   INDEX INTO THE BUFFER OF THE
  *   RECORD. IF THE BLOCK DOES NOT
  *   EXIST THEN ACCESS3 RETURNS
  *   THE INDEX INTO THE BUFFER OF
  *   THE RECORD AS IF THE BLOCK
  *   DID EXIST, AND SETS
  *   RECS*IN*BLK3 TO THAT VALUE.**
  *
  *
*********************************
*ITEM ACCESS3 INTEGER $         *
*** THE INDEX INTO F3*BUF OF    *
* THE RECORD**                  *
*ITEM BLOCK*NO INTEGER $        *
***THE FILE3 BLOCK CONTAINING   *
* THE RECORD**                  *
*ITEM REC*NO INTEGER $          *
***THE ABSOLUTE  NUMBER OF THE  *
* RECORD**                      *
*REMQUO(REC*NO, F3*BLKSIZ =     *
* BLOCK*NO, ACCESS3) $          *
*********************************
  *
  *
*********************************
*IF WRITE3 AND F3*BLK GQ 0 AND  *
* F3*BLK NQ BLOCK*NO $          *----+
*                               *    * ** MUST WRITE OUT F3*BLK
*********************************    *  BEFORE READING BLOCK*NO **
  *                                  *
  *                                  *
  *                                  *********************
```

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER
DESIGN DIAGRAM OF ACCESS3

```
                                     *TRANSFER'WRITE3 $ *
 *                                   ********************
 *
*********    *****************************
*IFEITH  *-+-*F3'BLK NQ BLOCK'NO AND     *
*********  * * BLOCK'NO LS F3'EMPTY $    *----+
 *         * *                           *    * '' BLOCK'NO EXISTS -- READ IT
 *         * *****************************    *  ''
 *         *                                  *
 *         *                                  *
 *         *                                 **************
 *         *                                 *IF DEBUG4 $ *   ********************************
 *         *                                 *''7''       *---*MESSAGE = 6H(READ3 ) $        *
 *         *                                 *            *   *T'MESS = IFORMAT(REC'NO) $    *
 *         *                                 **************   *MESSAGE = CAT(MESSAGE, T'MESS)*
 *         *                                  *               * $                            *
 *         *                                  *               *OUT(MESSAGE, MESS'SW) $       *
 *         *                                  *               ********************************
 *         *                                  *
 *         *                                 *********    ****************
 *         *                                 *IFEITH  *-+-*NEW'FILE3 $   *----+
 *         *                                 *********  * ****************    * '' LATEST VERSION OF FILE3 IS
 *         *                                  *         *                     *  FILE3'2 ''
 *         *                                  *         *                     *
 *         *                                  *         *
 *         *                                  *         *                    *****************************
 *         *                                  *         *                    *OPEN INPUT FILE3'2 $       *
 *         *                                  *         *                    *POS(FILE3'2) = BLOCK'NO $  *
 *         *                                  *         *                    *INPUT FILE3'2 RECS'IN'BLK3,*
 *         *                                  *         *                    * F3'BUF $                  *
 *         *                                  *         *                    *SHUT INPUT FILE3'2 $       *
 *         *                                  *         *                    *****************************
 *         *                                  *         *
 *         *                                  *         * ***********
 *         *                                  *         +-*ORIF 1 $ *----+
 *         *                                  *           ***********    * '' LATEST VERSION OF FILE3 IS
 *         *                                  *                          *  IN FILE3'1 ''
 *         *                                  *                          *
 *         *                                  *
 *         *                                  *                         *****************************
 *         *                                  *                         *OPEN INPUT FILE3'1 $       *
 *         *                                  *                         *POS(FILE3'1) = BLOCK'NO $  *
 *         *                                  *                         *INPUT FILE3'1 RECS'IN'BLK3,*
 *         *                                  *                         * F3'BUF $                  *
 *         *                                  *                         *SHUT INPUT FILE3'1 $       *
 *         *                                  *                         *****************************
 *         *                                  *
 *         *                                 ********************************
 *         *                                 *RECS'IN'BLK3 = MAX(RECS'IN'BLK*
 *         *                                 *3, ACCESS3) $                 *
 *         *                                 ********************************
 *         *
 *         * *****************************
 *         +-*ORIF F3'BLK NQ BLOCK'NO $  *----+
 *         * *****************************    * '' BLOCK'NO DOESN'T YET EXIST
 *         *                                  *
```

```
C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER                                                     PAGE 20
DESIGN DIAGRAM OF ACCESS3

 .         .                                 ***************************
 .         .                                 *RECS'IN'BLK3 = ACCESS3 $ *
 .         .                                 ***************************
 .         .
 .         .  ************
 .         +-*ORIF 1 $  *----+
 .            ************   .  ** BLOCK'NO IS ALREADY IN CORE
 .                           .  ..
 .                           .
 .                           .
 .                          ********************************
 .                          *RECS'IN'BLK3 = MAX(RECS'IN'BLK*
 .                          *3, ACCESS3) $                 *
 .                          ********************************
 .
**********************
*F3'BLK = BLOCK'NO $ *
*RETURN $            *
**********************
```

```
DESIGN DIAGRAM OF ACCESS4

**************************
*PROC ACCESS4(REC'NO) $  *
**************************
 .
 .
 .
 .  ** THE FUNCTION ACCESS4 IS THE
 .   FILE 4 INTERFACE. IT IS THE
 .   SAME AS ACCESS3 EXCEPT THAT
 .   IT DOESN'T CALL
 .   TRANSFER'WRITE4 IF READSW IS
 .   ON. **
 .
 .
*******************************
*ITEM ACCESS4 INTEGER $       *
*** THE INDEX INTO F4'BUF OF  *
* THE RECORD**                *
*ITEM BLOCK'NO INTEGER $      *
*** THE FILE4 BLOCK CONTAINING*
* THE RECORD**                *
*ITEM REC'NO INTEGER $        *
***THE ABSOLUTE NUMBER OF THE *
* RECORD**                    *
*REMQUO(REC'NO, F4'BLKSIZ =   *
* BLOCK'NO, ACCESS4) $        *
*******************************
 .
 .
********************************
*IF WRITE4 AND F4'BLK GQ 0 AND *
* F4'BLK NQ BLOCK'NO $         *----+
*                              *    . ** MUST WRITE OUT F4'BLK
********************************    .  BEFORE READING BLOCK'NO **
 .                                  .
 .                                  .
 .                                 ********************
 .                                 *TRANSFER'WRITE4 $ *
 .                                 ********************
 .
*********    ***************************
*IFEITH  *-+-*F4'BLK NQ BLOCK'NO AND   *
*********  . * BLOCK'NO LS F4'EMPTY $  *----+
 .         . *                         *    . ** BLOCK'NO EXISTS -- READ IT
 .         . ***************************    .  **
 .         .                                .
 .         .                                .
 .         .                               ***************
 .         .                               *IF DEBUG5 $  *   ********************************
 .         .                               ***7**        *---*MESSAGE = 6H(READ4 ) $        *
 .         .                               *             *   *T'MESS = IFORMAT(REC'NO) $    *
 .         .                               ***************   *MESSAGE = CAT(MESSAGE, T'MESS)*
 .         .                                .                * $                            *
 .         .                                .                *OUT(MESSAGE, MESS'SW) $       *
 .         .                                .                ********************************
 .         .                                .
```

DESIGN DIAGRAM OF ACCESS4

```
.         .                              *********     **************
.         .                              *IFEITH  *-+--*NEW*FILE4 $  *----+
.         .                              *********  .  **************     * ** LATEST VERSION OF FILE4 IS
.         .                               .         .                     *  FILE4*2 **
.         .                               .         .                     *
.         .                               .         .                    *****************************
.         .                               .         .                    *OPEN INPUT FILE4*2 $        *
.         .                               .         .                    *POS(FILE4*2) = BLOCK*NO $   *
.         .                               .         .                    *INPUT FILE4*2 RECS*IN*BLK4,*
.         .                               .         .                    * F4*BUF $                   *
.         .                               .         .                    *SHUT INPUT FILE4*2 $        *
.         .                               .         .                    *****************************
.         .                               .         .
.         .                               .         .  ***********
.         .                               .         +-*ORIF 1 $  *----+
.         .                               .            ***********    * ** LATEST VERSION OF FILE4 IS
.         .                               .                           *  IN FILE4*1 **
.         .                               .                           *
.         .                               .                          *****************************
.         .                               .                          *OPEN INPUT FILE4*1 $        *
.         .                               .                          *POS(FILE4*1) = BLOCK*NO $   *
.         .                               .                          *INPUT FILE4*1 RECS*IN*BLK4,*
.         .                               .                          * F4*BUF $                   *
.         .                               .                          *SHUT INPUT FILE4*1 $        *
.         .                               .                          *****************************
.         .                               .
.         .                              *********     ***********   ********************************    ******************
.         .                              *IFEITH  *-+--*READSW $  *---*IF ACCESS4 GR RECS*IN*BLK4 $  *---*ACCESS4 = - 1 $ *
.         .                              *********  .  ***********   ********************************    ******************
.         .                                         .
.         .                                         .  ***********   ********************************
.         .                                         +-*ORIF 1 $  *---*RECS*IN*BLK4 = MAX(RECS*IN*BLK*
.         .                                            ***********   *4, ACCESS4) $                 *
.         .                                                          ********************************
.         .  ***************************
.         +-*ORIF F4*BLK NQ BLOCK*NO $  *----+
.         .  ***************************     * ** BLOCK*NO DOESN*T YET EXIST
.         .                                  *  **
.         .                                  .
.         .                                  .
.         .                                 *********     ***********   ******************
.         .                                 *IFEITH  *-+--*READSW $  *---*ACCESS4 = - 1 $ *
.         .                                 *********  .  ***********   ******************
.         .                                            .
.         .                                            .  ***********   ***************************
.         .                                            +-*ORIF 1 $  *---*RECS*IN*BLK4 = ACCESS4 $ *
.         .                                               ***********   ***************************
.         .
.         .  ***********
.         +-*ORIF 1 $  *----+
.            ***********    * ** BLOCK*NO IS ALREADY IN CORE
```

```
C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER                                                          PAGE 23
DESIGN DIAGRAM OF ACCESS4

                             .  ..
 .                           .
 .                           .
 .                          **********   ************   ********************************   ******************
 .                          *IFEITH  *-+-*READSM $  *---*IF ACCESS4 GR RECS*IN*BLK4 $  *---*ACCESS4 = - 1 $ *
 .                          ********** . ************   ********************************   ******************
 .                                     .
 .                                     .
 .                                     . ************   ********************************
 .                                     +-*ORIF 1 $  *---*RECS*IN*BLK4 = MAX(RECS*IN*BLK*
 .                                       ************   *4, ACCESS4) $                 *
 .                                                      ********************************
 .
**********************
*F4*BLK = BLOCK*NO $ *
*RETURN $            *
**********************
```

```
DESIGN DIAGRAM OF ACCESS'OUT


****************************
*PROC ACCESS'OUT(REC'NO) $ *
****************************
 *
 *
 *
 * '' THE FUNCTION ACCESS'OUT IS
 *   THE INTERFACE FOR THE PUTOUT
 *   FILES.  ITS PURPOSE AND
 *   OPERATION ARE ANALAGOUS TO
 *   THOSE OF ACCESS3. ''
 *
 *
*********************************
*ITEM ACCESS'OUT INTEGER $      *
*'' INDEX INTO LINES'OUT OF THE*
* RECORD''                      *
*ITEM BLOCK'NO INTEGER $        *
*''THE 'BLOCK' CONTAINING THE   *
* RECORD''                      *
*ITEM REC'NO INTEGER $          *
*'' ABSOLUTE NUMBER OF THE      *
* RECORD IN PUTOUT''            *
*REMQUO(REC'NO, OUT'BUF'SIZE =  *
* BLOCK'NO, ACCESS'OUT) $       *
*********************************
 *
 *
****************************
*IF BLOCK'NO NQ OUT'BLK AND *
* OUT'BLK GQ 0 $            *----+
*                          *     * '' NEED TO WRITE OLD BLOCK
****************************     *  BEFORE GETTING NE4 ONE ''
 *                               *
 *                               *
 *                              ************************
 *                              *TRANSFER'WRITE'OUT $ *
 *                              ************************
 *
*********   ***************************
*IFEITH  *-+-*BLOCK'NO NQ OUT'BLK AND    *
*********  * * BLOCK'NO LQ MAX'FOUTPUT $ *----+
 *         * *                          *     * '' BLOCK'NO EXISTS -- GET IT
 *         * ***************************      *  ''
 *         *                                  *
 *         *                                  *
 *         *                                 **************
 *         *                                 *IF DEBUG6 $  *     ***********************************
 *         *                                 *''7''       *---*MESSAGE = 14H(ACCESSOUT GET ) *
 *         *                                 *            *     * $                             *
 *         *                                 **************     *T'MESS = IFORMAT(BLOCK'NO) $  *
 *         *                                  *                 *MESSAGE = CAT(MESSAGE, T'MESS)*
 *         *                                  *                 * $                             *
 *         *                                  *                 *OUT(MESSAGE, MESS'SW) $        *
 *         *                                  *                 ***********************************
 *         *                                  *
```

```
C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER                                                          PAGE 25
DESIGN DIAGRAM OF ACCESS'OUT

 .         .
 .         .                                  *********    ************
 .         .                                  *IFEITH  *-+--*NEW'OUT $  *----+
 .         .                                  *********  .  ************     * ** LATEST VERSION IS IN
 .         .                                             .                   .  PUTOUT'2 **
 .         .                                             .                   .
 .         .                                             .                  **************************
 .         .                                             .                  *OPEN INPUT PUTOUT'2 $     *
 .         .                                             .                  *POS(PUTOUT'2) = BLOCK'NO **
 .         .                                             .                  * OUT'BUF'SIZE $           *
 .         .                                             .                  **************************
 .         .                                             .                   .
 .         .                                             .                   .
 .         .                                             .                  ******************************
 .         .                                             .                  *FOR I = 0, 1, OUT'BUF'SIZE - 1 *    ******
 .         .                                             .                  * $                            *--* 27 *
 .         .                                             .                  *                            *    ******
 .         .                                             .                  ******************************
 .         .                                             .                   .
 .         .                                             .                   .
 .         .                                             .                  **********************
 .         .                                             .                  *SHUT INPUT PUTOUT'2 $ *
 .         .                                             .                  **********************
 .         .                                             .
 .         .                                             .  **********
 .         .                                             +--*ORIF 1 $  *----+
 .         .                                                **********      * ** LATEST VERSION IS IN
 .         .                                                                *  PUTOUT'1 **
 .         .                                                                .
 .         .                                                                .
 .         .                                                               **************************
 .         .                                                               *OPEN INPUT PUTOUT'1 $     *
 .         .                                                               *POS(PUTOUT'1) = BLOCK'NO **
 .         .                                                               * OUT'BUF'SIZE $           *
 .         .                                                               **************************
 .         .                                                                .
 .         .                                                                .
 .         .                                                               ******************************
 .         .                                                               *FOR I = 0, 1, OUT'BUF'SIZE - 1 *    ******
 .         .                                                               * $                            *--* 28 *
 .         .                                                               *                            *    ******
 .         .                                                               ******************************
 .         .                                                                .
 .         .                                                                .
 .         .                                                               **********************
 .         .                                                               *SHUT INPUT PUTOUT'1 $ *
 .         .                                                               **********************
 .         .
 .         .  ****************************
 .         +--*ORIF BLOCK'NO NQ OUT'BLK $  *-----+
 .            ****************************       * ** BLJCK DOESN'T EXIST --
 .                                               *  BUFFER BLANKED  **
 .                                               .
 .                                              ******************************
 .                                              *EXTRA'BLOCK = TRUE $          *
```

```
                                                  *** THERE*S AN EXTRA BLOCK IN*
 *                                                * CORE **                    *
 *                                                ******************************
 *
************************
*OUT*BLK = BLOCK*NO $ *
*RETURN $             *
************************
```

DESIGN DIAGRAM OF ACCESS'OUT

```
****************
* 27  FROM 25  *
****************
 *
 *
 *
********************************
*INPUT PUTOUT'2 OUT'LINE($I$) $*
*                              *
********************************
```

```
****************
* 28  FROM 25  *
****************
       .
       .
       .
**********************************
*INPUT PUTOUT'1 OUT'LINE($I$) $*
*                                *
**********************************
```

```
DESIGN DIAGRAM OF ADVANCE*PAGE

**********************
*PROC ADVANCE*PAGE $  *
**********************
 *
 *
 *
 * ** THE PROCEDURE ADVANCE*PAGE
 *  SETS LINE*NO TO THE NUMBER OF
 *  THE TOP LINE OF THE NEXT
 *  EMPTY PAGE . LINE*NO THEN
 *  BECOMES THE DISPLACEMENT USED
 *  FOR FINDING ABSOLUTE LINE
 *  NUMBERS FROM RELATIVE ONES.
 *  **
 *
 *
*******************************
*ITEM LINE INTEGER $          *
*** DISPLACEMENT OF LINE*NO ON*
* CURRENT PAGE**              *
*ITEM PAGE INTEGER $          *
***A DUMMY VARIABLE USED IN   *
* CALLING REMQUO**            *
*******************************
 *
 *
***************
*IF DEBUG1 $  *    ********************************
***7**        *---*T*MESS = 4H(ADV ) $            *
*             *    *MESSAGE = IFORMAT(LINE*NO) $  *
***************    *MESSAGE = CAT(T*MESS, MESSAGE)*
 *                 * $                            *
 *                 *OUT(MESSAGE, MESS*SW) $       *
 *                 ********************************
 *
********************    ******************************
*IF LINE*NO GR 0 $  *---*REMQUO(LINE*NO, PAGE*LNGTH =*
********************    * PAGE, LINE) $              *
 *                      ******************************
 *                       *
 *                       *
 *                      *****************    ********************************
 *                      *IF LINE NQ 0 $  *---*LINE*NO = LINE*NO + PAGE*LNGTH*
 *                      *****************    * - LINE $                     *
 *                                           ********************************
 *
********************
*DISP = LINE*NO $ *
*RETURN $         *
********************
```

DESIGN DIAGRAM OF BOX'MAP

```
****************************
*PROC BOX'MAP(STMT'TYPE) $ *
****************************
 .
 .
 .  ''THE FUNCTION BOX'MAP ACCEPTS
 .   STMT'TOKENS USED BY PHASE 1
 .   AND MAPS THEM ONTO STATEMENT
 .   TYPES USED BY PHASE2.''
 .
********************************
*ITEM BOX'MAP INTEGER $        *
*''THE PHASE2 STATEMENT TYPE'' *
*ITEM STMT'TYPE INTEGER $      *
*''THE INPUT PARAMETER''       *
********************************
 .
 .
*********  ******************   ****************
*IFEITH *-+-*STMT'TYPE EQ 1 $ *---*BOX'MAP = 1 $ *
*********  ******************   ****************
 .         .
 .         .
 .         . *********************    **********************
 .         +-*ORIF STMT'TYPE EQ 2 $ *---*BOX'MAP = COMMENT'1 $ *
 .         . *********************    **********************
 .         .
 .         .
 .         . *********************    *********   *****************    **********************
 .         +-*ORIF STMT'TYPE EQ 3 $ *---*IFEITH *-+-*DISPLAY'DELIM $ *---*BOX'MAP = COMMENT'1 $ *
 .         . *********************    *********  . *****************    **********************
 .         .                                     .
 .         .                                     .
 .         .                                     . **********   ***************************
 .         .                                     +-*ORIF 1 $ *---*BOX'MAP = DELIM'COMMENT $ *
 .         .                                       **********   ***************************
 .         .
 .         .
 .         . *********************    *********   *****************    *****************
 .         +-*ORIF STMT'TYPE LQ 7 $ *---*IFEITH *-+-*DISPLAY'DELIM $ *---*BOX'MAP = SEQ $ *
 .         . *********************    *********  . *****************    *****************
 .         .                                     .
 .         .                                     .
 .         .                                     . **********   *************************
 .         .                                     +-*ORIF 1 $ *---*BOX'MAP = BLOCK'DELIM $ *
 .         .                                       **********   *************************
 .         .
 .         .
 .         . *********************    **********************
 .         +-*ORIF STMT'TYPE EQ 8 $ *---*BOX'MAP = PGM'TAIL $ *
 .         . *********************    **********************
 .         .
 .         .
 .         . **********************    *****************
 .         +-*ORIF STMT'TYPE LQ 37 $ *---*BOX'MAP = SEQ $ *
 .         . **********************    *****************
```

```
C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER
DESIGN DIAGRAM OF BOX'MAP

 *        * *************************   ************************
 *        +-*ORIF STMT'TYPE EQ 38 $ *---*BOX'MAP = COMMENT'2 $ *
 *        * *************************   ************************
 *        *
 *        *
 *        * *************************   ************************
 *        +-*ORIF STMT'TYPE LQ 41 $ *---*BOX'MAP = CONTROL'1 $ *
 *        * *************************   ************************
 *        *
 *        *
 *        * *************************   ************************
 *        +-*ORIF STMT'TYPE LQ 45 $ *---*BOX'MAP = CONTROL'2 $ *
 *        * *************************   ************************
 *        *
 *        *
 *        * *************************   ************************
 *        +-*ORIF STMT'TYPE LQ 48 $ *---*BOX'MAP = CONTROL'3 $ *
 *        * *************************   ************************
 *        *
 *        *
 *        * ***********   ************************
 *        +-*ORIF 1 $ *---*BOX'MAP = CONTROL'4 $ *
 *          ***********   ************************
 *
***********
*RETURN $ *
***********
```

DESIGN DIAGRAM OF BYTE'EM

```
*****************
*PROC BYTE'EM $  *
*****************
 *
 *
 *
 * ''THE PROCEDURE BYTE'EM READS
 *   THE DIAGRAM FROM EITHER
 *   PUTOUT1 OR PUTOUT2, TRUNCATES
 *   IT AND WRITES IT TO
 *   FINAL'OUT.''
 *
 *
********************************
*ITEM LINE'COUNT INTEGER $     *
*''NUMBER OF LINES IN THIS     *
* OUTPUT GROUP''               *
*ITEM PAGE INTEGER $           *
*''DUMMY USED IN CALLING       *
* REMQUO''                     *
*ITEM TEMPH1 CHARACTER $       *
*'' A TEMPORARY CHARACTER      *
* VARIABLE''                   *
*PAGE'WIDTH = PAGE'WIDTH + 1 $ *
*OPEN OUTPUT FINAL'OUT $       *
********************************
 *
 *
*********    **********************   *****************************   ********************************
*IFEITH  *-+-*MAX'FOUTPUT LS 0 $  *---*FOR I = 0, 1, LAST'LINE $  *---*OUTPUT FINAL'OUT BYTE($0,     *
*********  * **********************   *****************************   * PAGE'WIDTH$)(OUT'LINE($I$)) $*
           *                                                          ********************************
           *
           * ************   ********************   *******************************
           +-*ORIF 1 $  *---*IF EXTRA'BLOCK $  *---*MAX'FOUTPUT = MAX'FOUTPUT + 1*
             ************   ********************   * $                           *
                             *                     *******************************
                             *
                            *********    ************   ************************
                            *IFEITH  *-+-*NEW'OUT $  *---*OPEN INPUT PUTOUT'2 $ *
                            *********  * ************   ************************
                                       *                 *
                                       *                 *
                                       *                ********************************   ******
                                       *                *FOR J = 0, 1, MAX'FOUTPUT $  *---* 33 *
                                       *                ********************************   ******
                                       *
                                       * ************   ************************
                                       +-*ORIF 1 $  *---*OPEN INPUT PUTOUT'1 $ *
                                         ************   ************************
                                                         *
                                                         *
                                                        ********************************   ******
                                                        *FOR J = 0, 1, MAX'FOUTPUT $  *---* 34 *
                                                        ********************************   ******
```

```
****************
* 33  FROM 32  *
****************
 *
 *
 *
*********    **********************    ********************************
*IFEITH  *-+-*J EQ MAX'FOUTPUT $  *---*REMQUO(LAST'LINE, OUT'BUF'SIZE*
*********  * **********************    * = PAGE, LINE'CJUNT) $        *
 *         *                           ********************************
 *         *
 *         * ************    *******************************
 *         +-*ORIF 1 $  *---*LINE'COUNT = OUT'BUF'SIZE - 1*
 *           ************    * $                           *
 *                           *'' COUNT FOR OTHER BLOCKS''  *
 *                           *******************************
 *
*********    *****************    ******************************    ********************************
*IFEITH  *-+-*J EQ OUT'BLK $  *---*FOR I = 0, 1, LINE'COUNT $  *---*OUTPUT FINAL'OUT BYTE($0,     *
*********  * *****************    ******************************    * PAGE'WIDTH$)(OUT'LINE($I$)) $*
           *                                                        ********************************
           *
           * ************    *********************
           +-*ORIF 1 $  *---*POS(PUTOUT'2) = J **
             ************    * OUT'BUF'SIZE $    *
                             *********************
                              *
                              *
                            ******************************    ***************************
                            *FOR I = C, 1, LINE'COUNT $  *---*INPUT PUTOUT'2 TEMPH1 $  *
                            ******************************    *OUTPUT FINAL'OUT BYTE($0,*
                                                              * PAGE'WIDTH$)(TEMPH1) $  *
                                                              ***************************
```

```
DESIGN DIAGRAM OF BYTE*EM

****************
* 34  FROM 32  *
****************
 .
 .
 .
*********   **********************   ********************************
*IFEITH *-+-*J EQ MAX*FOUTPUT $  *---*REMQUO(LAST*LINE, OUT*BUF*SIZE*
********* . **********************   * = PAGE, LINE*CJUNT) $        *
 .        .                          ********************************
 .        .
 .        . ************   *******************************
 .        +-*ORIF 1 $  *---*LINE*COUNT = OUT*BUF*SIZE - 1*
 .          ************   * $                           *
 .                         *******************************
 .
*********   *****************   *****************************   ********************************
*IFEITH *-+-*J EQ OUT*BLK $  *---*FOR I = 0, 1, LINE*COUNT $  *---*OUTPUT FINAL*OUT BYTE($0,     *
********* . *****************   *****************************   * PAGE*WIDTH$)(OUT*LINE($I$)) $*
          .                                                     ********************************
          .
          . ************   ***********************
          +-*ORIF 1 $  *---*POS(PUTOUT*1) = J **
            ************   * OUT*BUF*SIZE $     *
                           ***********************
                              .
                              .
                           *****************************   ***************************
                           *FOR I = 0, 1, LINE*COUNT $  *---*INPUT PUTOUT*1 TEMPH1 $  *
                           *****************************   *OUTPUT FINAL*OUT BYTE($0,*
                                                           * PAGE*WIDTH$)(TEMPH1) $  *
                                                           ***************************
```

```
C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER                                                    PAGE 35
DESIGN DIAGRAM OF CLOSE*REC

*****************************
*PROC CLOSE*REC(TERM*REC) $  *
*****************************
 *
 *
 *
 * ** THE PROCEDURE CLOSE*REC
 *   ASSIGNS THE LINES AND
 *   BLOCK*WIDTH FIELDS OF FILE 3
 *   RECORDS. IT CALLS PLACE,
 *   WHICH COMPLETES THE PART 1
 *   PROCESSING OF THE FILE 3
 *   RECORD. **
 *
 *
********************************
*ITEM STMT*TYPE INTEGER $      *
*** THE PART 2 STMT TYPE OF THE*
* RECORD**                     *
*ITEM TERM*REC INTEGER $       *
***RECORD NUMBER OF THE RECORD *
* BEING CLOSED**               *
*ITEM TEMPI1 INTEGER $         *
***TEMPORARY INTEGER **        *
*ITEM TEMPI2 INTEGER $         *
***TEMPORARY INTEGER **        *
*STMT*TYPE =(/ STMT*UNIT($ACCES*
*S3(TERM*REC)$) /) $           *
********************************
 *
 *
 *
 * **INDICATE THAT THIS RECORD
 *  HAS BEEN CLOSED **
 *
 *
********************************
*STMT*UNIT($ACCESS3(TERM*REC)$)*
* = - STMT*TYPE $              *
********************************
 *
 *
*******************************
*IF STMT*TYPE GQ CONTROL*1 AND *
* SINGLE*SPACE $               *-----*
*                              *     * ** HAVE A CONTROL BOX -- IT
*******************************      *  MUST HAVE AN ODD NUMBER OF
 *                                   *  LINES **
 *                                   *
 *                                   *
 *                                  *******************************
 *                                  *REMQUO(LINES($ACCESS3(TERM*REC*
 *                                  *)$), 2 = TEMPI1, TEMPI2) $    *
 *                                  *******************************
 *                                   *
 *                                   *
```

```
DESIGN DIAGRAM OF CLOSE'REC

 *
 *                                    ******************
 *                                    *IF TEMPI2 EQ 0 $ *----+
 *                                    ******************     * ** ADD A LINE **
 *                                                           *
 *                                                           *
 *                                                           ******************************
 *                                                           *LINES($ACCESS3(TERM'REC)$) = *
 *                                                           * LINES($ACCESS3(TERM'REC)$) +*
 *                                                           * 1 $                        *
 *                                                           ******************************
 *
 *
 *
 * **ADD LINES FOR BOX EDGES AND
 *   DOUBLE SPACE OPTION **
 *
 *
*********    *****************    ******************************
*IFEITH *-+-*SINGLE'SPACE $ *---*LINES($ACCESS3(TERM'REC)$) = *
*********  *  *****************    * LINES($ACCESS3(TERM'REC)$) +*
 *         *                      * 2 $                        *
 *         *                      ******************************
 *         *
 *         *
 *          ******************       ******************************
 *         +-*ORIF DOUBLE'SPACE $ *---*LINES($ACCESS3(TERM'REC)$) = 2*
 *           ******************       * * LINES($ACCESS3(TERM'REC)$) *
 *                                    * + 1 $                        *
 *                                    ******************************
*******************
*WRITE3 = TRUE $  *
*PLACE(TERM'REC) $ *
*******************
 *
 *
*****************    **************************
*IF STUMP'FOUND $ *---*RESOLVE'STUMP(TERM'REC) $ *
*****************    **************************
 *
 *
**********
*RETURN $ *
**********
```

DESIGN DIAGRAM OF COMPUTE'PAGE'NUMBERS

```
******************************
*PROC COMPUTE'PAGE'NUMBERS $  *
******************************
 *
 *
 *
 * ''THE PROCEDURE
 *   COMPUTE'PAGE'NUMBERS ASSIGNS
 *   REFEPENCE NUMBERS TO STUMPS
 *   AND IF HEADING AND
 *   TABLE'OF'CONTENTS ARE ON,
 *   CALLS GENERATE'HEADER'ENTRY
 *   TO ENTER A MODULE NAME INTO
 *   THE TABLE OF CONTENTS. ''
 *
 *
********************************
*ITEM ABS'PAGE'NO INTEGER $    *
*''NUMBER OF PAGE ON WHICH     *
* MODULES START'' ''NUMBER OF P*
*ITEM CONTENTS'PAGES INTEGER $ *
*''PAGES SPANNED BY CONTENTS   *
* TABLE''                      *
*ITEM CUM'PAGES INTEGER $      *
*''CUMULATIVE PAGE COUNT''     *
*ITEM STUMP'COUNT INTEGER P 0 $*
*                              *
*''CUMULATIVE STUMP COUNT ''   *
********************************
 *
 *
***************
*IF DEBUG1 $  *    ********************************
*''5''        *---*MESSAGE = 11H(COMPUTE NOS) $ *
*             *    *OUT(MESSAGE, MESS'SW) $      *
***************    ********************************
 *
 *
************************
*CUM'PAGES = PAGE'NO $ *
************************
 *
 *
********************************
*IF HEADING AND TABLE'OF'CONTEN *    ********************************
*TS $                           *---*CONTENTS'PAGES = PROC'COUNT /*
*                               *    * (PAGE'LNGTH - HEAD'NO) + 1 $*
********************************     *CUM'PAGES = CUM'PAGES +      *
 *                                   * CONTENTS'PAGES $             *
 *                                   *GENERATE'CONTENTS'HEADER $    *
 *                                   ********************************
 *
**************************
*CUR'GROUP = NEXT(30$) $ *
**************************
 *
```

DESIGN DIAGRAM OF COMPUTE*PAGE*NUMBERS

```
****************************    *********    ************   *********************************
*DO WHILE [CUR*GROUP GR 0]  *---*IFEITH  *-+-*HEADING $ *---*ABS*PAGE*NO = CUM*PAGES + 1 $  *
****************************    *********  . ************   *CUM*PAGES = CUM*PAGES +        *
 .                               .         .                * PAGE*REF($CUR*GROUP$) $       *
 .                               .         .                *PAGE*REF($CUR*GROUP$) =        *
 .                               .         .                * ABS*PAGE*NO $                 *
 .                               .         .                *********************************
 .                               .         .                 .
 .                               .         .                 .
 .                               .         .                *****************************
 .                               .         .                *IF TABLE*OF*CONTENTS AND   *    ****************************
 .                               .         .                * PROC*NAME($CUR*GROUP$) NQ *---*GENERATE*CONTENTS*ENTRY $ *
 .                               .         .                * SPACES(MAXCOL) $          *    ****************************
 .                               .         .                *****************************
 .                               .         .
 .                               .         . ********************************
 .                               .         +-*ORIF PROC*NAME($CUR*GROUP$) EQ*
 .                               .           * SPACES(MAXCOL) $             *----+
 .                               .           *                              *    . ** NUMBER STUMPS ACCORDING TO
 .                               .           ********************************    .  ORDER OF APPEARANCE **
 .                               .                                               . ** RATHER THAN BY PAGE NUMBER
 .                               .                                               .  **
 .                               .                                               .
 .                               .                                               .
 .                               .                                               *******************************
 .                               .                                               *STUMP*COUNT = STUMP*COUNT + 1*
 .                               .                                               * $                           *
 .                               .                                               *PAGE*REF($CUR*GROUP$) =      *
 .                               .                                               * STUMP*COUNT $               *
 .                               .                                               *******************************
 .                               .                                                .
 .                               .                                                .
 .                               .                                               ***************
 .                               .                                               *IF DEBUG25 $ *    ******
 .                               .                                               ***7**        *---* 39 *
 .                               .                                               *             *    ******
 .                               .                                               ***************
 .                               .
 .                              *******************************
 .                              *CUR*GROUP = NEXT($CUR*GROUP$)*
 .                              * $                           *
 .                              *******************************
 .
***********
*RETURN $ *
***********
```

```
C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER                                                     PAGE 39
DESIGN DIAGRAM OF COMPUTE*PAGE*NUMBERS

****************
* 39  FROM 38  *
****************
 *
 *
 *
*********************************
*MESSAGE = 8H(STUMPNO ) $       *
*T*MESS = IFORMAT(STUMP*COUNT)  *
* $                             *
*MESSAGE = CAT(MESSAGE, T*MESS) *
* $                             *
*OUT(MESSAGE, MESS*SW) $        *
*********************************
```

DESIGN DIAGRAM OF CONNECT'BOXES

```
***********************
*PROC CONNECT'BOXES $  *
***********************
 *
 *
 *
 * '' THE PROCEDURE CONNECT'BOXES
 *   DETERMINES HOW TO CONNECT A
 *   CODE BLOCK TO ITS FATHER (IF
 *   IT HAS ONE). IF THE CODE
 *   BLOCK IS A STUMP THEN
 *   CONNECT'BOXES DRAWS THE STUMP
 *   REFERENCE DISPLAY BOX FOR THE
 *   STUMP. ''
 *
 *
**********************************
*ITEM DISPLAY'TOP INTEGER $      *
*'' TOP LINE OF THE DISPLAY      *
* BOX''                          *
*ITEM FATHER INTEGER $           *
*''RECORD NUMBER OF CODE         *
* BLOCK'S FATHER''               *
*ITEM FATHER'BOTTOM INTEGER $    *
*''FATHER'S BOTTOM LINE''        *
*ITEM FATHER'MID INTEGER $       *
*''FATHER'S MIDPOINT''           *
*ITEM FATHER'TOP INTEGER $       *
*''FATHER'S TOP LINE''           *
*ITEM FATHER'TYPE INTEGER $      *
*''FATHER'S TYPE ''              *
*ITEM HORIZ B $                  *
*'' TYPE OF CONNECTION. HORIZ    *
* OR VERT''                      *
*ITEM LINE INTEGER $             *
*''LINE'NO'S DISPLACEMENT ON     *
* CURRENT PAGE''                 *
*ITEM LNGTH INTEGER $            *
*''USED TO STORE LENGTH OF       *
* STRINGS''                      *
*ITEM PAGE INTEGER $             *
*''DUMMY USED TO CALL REMQUO ''  *
*                                *
*ITEM X1 INTEGER $               *
*''X COORD OF CONNECTING LINE'S  *
* START''                        *
*ITEM X2 INTEGER $               *
*'' X COORD OF CONNECTING        *
* LINE'S END''                   *
*ITEM Y1 INTEGER $               *
*'' Y COORD OF CONNECTING        *
* LINE'S START ''                *
*ITEM Y2 INTEGER $               *
*'' Y COORD OF CONNECTING        *
* LINE'S END''                   *
**********************************
```

```
C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER                                                          PAGE 41
DESIGN DIAGRAM OF CONNECT'BOXES

**********   ******************    **********************
*IFEITH  *-+-*H'FATHER NQ 0 $  *---*FATHER = H'FATHER $ *
********** * ******************    *HORIZ = TRUE $      *
 *         *                       **********************
 *         *
 *         * ***********************    **********************
 *         +-*ORIF V'FATHER NQ 0 $  *---*FATHER = V'FATHER $ *
 *         * ***********************    *HORIZ = FALSE $     *
 *         *                            **********************
 *         *
 *         * ***********
 *         +-*ORIF 1 $  *----+
 *           ***********     * ''NOTHING TO CONNECT ''
 *                           *
 *                           *
 *                          ***********
 *                          *RETURN $ *
 *                          ***********
 *
**************
*IF DEBUG8 $ *     ********************************
*''11''       *---*MESSAGE = 8H(CONNECT ) $      *
*            *     *T'MESS = IFORMAT(FATHER) $    *
**************     *MESSAGE = CAT(MESSAGE, T'MESS)*
 *                 * $                            *
 *                 *T'MESS = 4H( TO ) $           *
 *                 *MESSAGE = CAT(MESSAGE, T'MESS)*
 *                 * $                            *
 *                 *T'MESS = IFORMAT(CUR'REC) $   *
 *                 *MESSAGE = CAT(MESSAGE, T'MESS)*
 *                 * $                            *
 *                 *OUT(MESSAGE, MESS'SW) $       *
 *                 ********************************
 *
**********   ****************
*IFEITH  *-+-*FATHER LS 0 $  *----+
********** * ****************     * '' IT'S A STUMP ROOT ''
 *         *                      *
 *         *                      *
 *         *                      ******************************
 *         *                      *REMQUO(LINE'NO, PAGE'LNGTH =*
 *         *                      * PAGE, LINE) $              *
 *         *                      ******************************
 *         *                      *
 *         *                      **************    ********************************
 *         *                      *IF HEADING $ *---*PAGE'REF($CUR'GROUP$) = PAGE +*
 *         *                      **************    * 1 $                          *
 *         *                      *                 ********************************
 *         *                      *
 *         *                      **********   ********************************    **************
 *         *                      *IFEITH  *-+-*PAGE'RE=($CUR'GROUP$) LS 10 $  *---*LNGTH = 1 $ *
 *         *                      ********** * ********************************    **************
 *         *                      *          *
 *         *                      *          *
 .         .                      .         . **********************************
 .         .                      .         +-*ORIF PAGE'REF($CUR'GROUP$) LS *    **************
 .         .                      .         . * 100 $                        *---*LNGTH = 2 $ *
 .         .                      .         . *                           *    **************
 .         .                      .         . **********************************
 .         .                      .         .
 .         .                      .         .
 .         .                      .         . **********************************
 .         .                      .         +-*ORIF PAGE'REF($CUR'GROUP$) LS *    **************
 .         .                      .         . * 1000 $                       *---*LNGTH = 3 $ *
 .         .                      .         . *                           *    **************
 .         .                      .         . **********************************
 .         .                      .         .
 .         .                      .         .
 .         .                      .         . ***********   ************************
 .         .                      .         +-*ORIF 1 $ *---*PH2ERR(6H(WHOOPS)) $ *
 .         .                      .           ***********   ************************
 .         .                      .
 .         .                     *******************************
 .         .                     *DISPLAY'TOP = TOP'LINE -     *
 .         .                     * V'SPACE - 3 $               *
 .         .                     *STARS(MARGIN, DISPLAY'WIDTH,*
 .         .                     * DISPLAY'TOP) $              *
 .         .                     *******************************
 .         .                      .
 .         .                      .
 .         .                     *********  ***********    ************************************
 .         .                     *IFEITH  *-+-*HEADING $  *---*TEMPC = 16H(*                  *)*
 .         .                     *********  * ***********    * $                               *
 .         .                      .         .                ************************************
 .         .                      .         .
 .         .                      .         . ***********   ************************
 .         .                      .         +-*ORIF 1 $ *---*TEMPC = 6H(*    *) $ *
 .         .                      .           ***********   ************************
 .         .                      .
 .         .                     *********************************
 .         .                     *BYTE($MARGIN, DISPLAY'WIDTH$)(*
 .         .                     *OUT'LINE($ACCESS'OUT(DISPLAY'T*
 .         .                     *OP + 1)$)) = TEMPC $           *
 .         .                     *TEMPC = IFORMAT(PAGE'REF($CUR'*
 .         .                     *GROUP$)) $                     *
 .         .                     *BYTE($MARGIN + 2,              *
 .         .                     * LNGTH$)(OUT'LINE($ACCESS'OUT(*
 .         .                     *DISPLAY'TOP + 1)$)) = BYTE($6,*
 .         .                     * LNGTH$)(TEMPC) $              *
 .         .                     *********************************
 .         .                      .
 .         .                      .
 .         .                     ***************    **********************************
 .         .                     *IF HEADING $  *---*BYTE($MARGIN + 6,                *
 .         .                     ***************    * 4$)(OUT'LINE($ACCESS'OUT(DISP*
 .         .                      .                 *LAY'TOP + 1)$)) = 4H(FROM) $  *
 .         .                      .                 *TEMPC = IFORMAT(FROM'PAGE($CUR*
 .         .                      .                 *'GROUP$)) $                    *
 .         .                      .                 *LNGTH = LENGTH(TEMPC) $        *
 .         .                      .                 *BYTE($MARGIN + 11,             *
                                                  * LNGTHS) (OUT*LINE($ACCESS*OUT(*
.         .                     .                 *DISPLAY*TOP + 1)$)) = BYTE($6,*
.         .                     .                 * LNGTHS)(TEMPC) $             *
.         .                     .                 ********************************
.         .                     .
.         .                    ******************************
.         .                    *STARS(MARGIN, DISPLAY*WIDTH,*
.         .                    * DISPLAY*TOP + 2) $         *
.         .                    *X1 = MARGIN + 1 $           *
.         .                    *Y1 = DISPLAY*TOP + 2 $      *
.         .                    *X2 = MARGIN + 1 $           *
.         .                    *Y2 = TOP*LINE $             *
.         .                    ******************************
.         .
.         . ************   ********************************
.         +-*ORIF 1 $  *---*FATHER*TOP = START*LINE($ACCES*
.           ************   *S3(FATHER)$) + DISP $         *
.                          *FATHER*BOTTOM =               *
.                          * STOP*LINE($ACCESS3(FATHER)$) *
.                          * + DISP $                     *
.                          *FATHER*MID = MIDPT($ACCESS3(FA*
.                          *THER)$) + DISP $              *
.                          *FATHER*TYPE =(/               *
.                          * STMT*UNIT($ACCESS3(FATHER)$) *
.                          * /) $                         *
.                          ********************************
.                           .
.                           .
.                          *********   ***********   ********************************
.                          *IFEITH *-+-*HORIZ $  *---*X1 = STOP*COL($ACCESS3(FATHER)*
.                          ********* . ***********   *$) $                          *
.                                    .               *Y1 = FATHER*MID $             *
.                                    .               *X2 = LEFT*COL $               *
.                                    .               ********************************
.                                    .                .
.                                    .                .
.                                    .               *********   ****************************   ******************
.                                    .               *IFEITH *-+-*STMT*TYPE EQ COMMENT*2 $  *---*Y2 = TOP*LINE $ *
.                                    .               ********* . ****************************   ******************
.                                    .                         .
.                                    .                         .
.                                    .                         . ************   **********************
.                                    .                         +-*ORIF 1 $  *---*Y2 = TOP*LINE + 1 $ *
.                                    .                           ************   **********************
.                                    . ************   *********   ******************************  ******
.                                    +-*ORIF 1 $  *---*IFEITH *-+-*FATHER*TYPE EQ CONTROL*3 $  *--* 45 *
.                                      ************   ********* . ******************************  ******
.                                                      .        .
.                                                      .        .
.                                                      .        . ************   ********************************
.                                                      .        +-*ORIF 1 $  *---*X1 = START*COL($ACCESS3(FATHER*
.                                                      .          ************   *)$) + 1 $ X1 = START*COL($ACCE*
.                                                      .                         *Y1 = FATHER*BOTTOM $          *
.                                                      .                         ********************************
.                                                      .
.                                                      .
```

```
C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER                                                          PAGE 44
DESIGN DIAGRAM OF CONNECT'BOXES

 *
 *                                                      **********   ****************************   **********************
 *                                                      *IFEITH  *-+-*STMT'TYPE EQ CONTROL'3 $  *---*X2 = LEFT'COL $      *
 *                                                      ********** * ****************************   *Y2 = TOP'LINE + 1 $ *
 *                                                                 *                                **********************
 *                                                                 *
 *                                                                 * ************   **********************
 *                                                                 +-*ORIF 1 $  *---*X2 = LEFT'COL + 1 $ *
 *                                                                   ************   *Y2 = TOP'LINE $     *
 *                                                                                  **********************
******************************
*DRAW'LINE(X1, Y1, X2, Y2) $ *
*RETURN $                    *
******************************
```

```
DESIGN DIAGRAM OF CONNECT'BOXES

****************
* 45  FROM 43  *
****************
 .
 .
 .
********************************
*X1 = START'COL($ACCESS3(FATHER*
*)$) $                         *
*Y1 = FATHER'TOP + 1 $         *
********************************
```

```
DESIGN DIAGRAM OF CONSTRUCT'LINE

*************************
*PROC CONSTRUCT'LINE $  *
*************************
 *
 *
 *
 * '' THE PROCEDURE CONSTRUCT
 *   LINE EMBEDS THE LINE OF INPUT
 *   TEXT PROVIDED BY THE
 *   PROCEDURE EXTRACT'TEXT IN THE
 *   SIDES OF THE OF THE BOX IN
 *   WHICH IT WILL APPEAR.
 *   CONSTRUCT'LINE ALSO IS
 *   RESPONSIBLE FOR OUTPUTTING
 *   BLANK TEXT LINES IF THE
 *   DOUBLE'SPACE OPTIONS IS IN
 *   EFFECT. DOUBLE SPACING IS
 *   HANDLED BY ITERATING THROUGH
 *   THE ROUTINE INC TIMES WHERE
 *   INC IS SET ACCORDING TO THE
 *   VALUE OF SKIP. THE BLANK LINE
 *   FLAG . ''
 *
 *
**********************************
*ITEM INC INTEGER $              *
*''THE UPPPER BOUND FOR THE      *
* OUTPUT LOOP''                  *
*ITEM LINE INTEGER $             *
*''LINE'NO'S DISPLACEMENT ON     *
* CURRENT PAGE''                 *
*ITEM LNGTH INTEGER $            *
*''USED TO STORE STRING          *
* LENGTHS''                      *
*ITEM PAGE INTEGER $             *
*''DUMMY VARIABLE USED FOR CALL*
* TO REMQUO''                    *
*ITEM REAL'START INTEGER $       *
*''STORES THE COLUMN IN WHICH    *
* TEXT STARTS''                  *
**********************************
 *
 *
***************
*IF DEBUG9 $  *     ******************************
*''5''        *---*MESSAGE = 9H(CONSTRUCT) $ *
*             *     *OUT(MESSAGE, MESS'SW) $   *
***************     ******************************
 *
 *
******************************
*LNGTH = LENGTH(NEW'TEXT) $ *
******************************
 *
..*
 *
```

```
C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER
DESIGN DIAGRAM OF CONSTRUCT'LINE

**********   *********   ************
*IFEITH  *-+-*SKIP $  *---*INC = 1 $ *
********** . *********   ************
 .         .
 .         .
 .         . ***********   ************
 .         +-*ORIF 1 $  *---*INC = 0 $ *
 .           ***********   ************
 .
**********************   ****************************
*FOR I = 0, 1, INC $  *---*LINE'NO = LINE'NO + I $ *
**********************   ****************************
 .                         .
 .                         .
 .                         ****************
 .                         *IF HEADING $  *----+
 .                         ****************    . ''MAKE SURE WE'RE NOT IN
 .                         .                   .  HEADING ''
 .                         .                   .
 .                         .                   .
 .                         .                   ******************************
 .                         .                   *REMQUO(LINE'NO, PAGE'LNGTH =*
 .                         .                   * PAGE, LINE) $              *
 .                         .                   ******************************
 .                         .                   .
 .                         .                   .
 .                         .                   ************************
 .                         .                   *IF LINE LQ HEADROOM $ *----+
 .                         .                   ************************    . '' WE ARE -- INCREMENT LINE'NO
 .                         .                                               .  ''
 .                         .                                               .
 .                         .                                               .
 .                         .                                               ******************************
 .                         .                                               *OUTPUT'HEADER(LINE'NO - LINE)*
 .                         .                                               * $                           *
 .                         .                                               ******************************
 .                         .
 .                         ********************************
 .                         *TEMPC = OUT'LINE($ACCESS'OUT(L*
 .                         *INE'NO)$) $                   *
 .                         ********************************
 .                          .
 .                          .
 .                         **********   ***************************     *********************************
 .                         *IFEITH  *-+-*STMT'TYPE EQ COMMENT'2 $  *---*BYTE($LEFT'COL + 1, 2$)(TEMPC)*
 .                         ********** . ***************************     * = 2H('  ) $                   *
 .                          .         .                                 *REAL'START = LEFT'COL + 3 $    *
 .                          .         .                                 *********************************
 .                          .         .
 .                          .         . ***************************
 .                          .         +-*ORIF STMT'TYPE EQ SEQ OR  *     *********************************
 .                          .         . * STMT'TYPE EQ >GM'TAIL OR *---*BYTE($LEFT'COL, 1$)(TEMPC) =  *
 .                          .         . * STMT'TYPE EQ 0 $         *     * 1H(*) $                       *
 .                          .         . ***************************     *BYTE($RIGHT'COL, 1$)(TEMPC) = *
 .                          .         .                                 * 1H(*) $                       *
 .                          .         .                                 *REAL'START = LEFT'COL + 1 $    *
```

DESIGN DIAGRAM OF CONSTRUCT'LINE

```
                                                                        *******************************
 .                         .          .
 .                         .          . ********************************   **********   *************  ******
 .                         .          +-*ORIF STMT'TYPE GQ CONTROL'1 $ *---*IFEITH  *-+-*PAST'MID $ *--* 49 *
 .                         .            ********************************   ********** . *************  ******
 .                         .                                               .          .
 .                         .                                               .          .
 .                         .                                               .          . ***********  ******
 .                         .                                               .          +-*ORIF 1 $ *--* 50 *
 .                         .                                               .            ***********  ******
 .                         .                                               .
 .                         .                                               **********   ***************************  ******
 .                         .                                               *IFEITH  *-+-*STMT'TYPE EQ CONTROL'3 $ *--* 51 *
 .                         .                                               ********** . ***************************  ******
 .                         .                                               .          .
 .                         .                                               .          .
 .                         .                                               .          . ***********  ******
 .                         .                                               .          +-*ORIF 1 $ *--* 52 *
 .                         .                                               .            ***********  ******
 .                         .                                               .
 .                         .                                               ***************************
 .                         .                                               *BYTE($MIN(RIGHT'POS,     *
 .                         .                                               * RIGHT'COL), 1$)(TEMPC) =*
 .                         .                                               * 1H(*) $                 *
 .                         .                                               ***************************
 .                         .                                               .
 .                         .                                               .
 .                         .                                               ***************************   ********************
 .                         .                                               *IF LINE'NO EQ MIDPOINT $ *---*PAST'MID = TRUE $ *
 .                         .                                               ***************************   ********************
 .                         .
 .                        *******************************   ******************************
 .                        *IF I EQ INC AND LNGTH GR 0 $ *---*LNGTH = MIN(LNGTH, WIDTH) $ *
 .                        *******************************   *BYTE($REAL'START,           *
 .                         .                                * LNGTH$)(TEMPC) = BYTE($6,  *
 .                         .                                * LNGTH$)(NEW'TEXT) $        *
 .                         .                                ******************************
 .                         .
 .                        ********************************
 .                        *OUT'LINE($ACCESS'OUT(LINE'NO)$*
 .                        *) = TEMPC $                   *
 .                        ********************************
 .                         .
 .                         .
 .                        **************
 .                        *IF DEBUG9 $ *    ***********************
 .                        *''2''        *---*OUT(TEMPC, MESS'SW) $ *
 .                        *            *    ***********************
 .                        **************
 .
***********
*RETURN $ *
***********
```

```
****************
* 49  FROM 48  *
****************
 *
 *
 *
*******************************
*RIGHT'POS = RIGHT'POS - 1 $ *
*******************************
```

```
****************
* 50  FROM 48  *
****************
 *
 *
 *
******************************
*RIGHT'POS = RIGHT'POS + 1 $ *
******************************
```

```
C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER
DESIGN DIAGRAM OF CONSTRUCT'LINE

****************
* 51  FROM 48  *
****************
 *
 *
 *
*******************************    *******************************
*IF LINE'NO EQ TOP'LINE + 1 $  *---*BYTE($LEFT'COL, 1$)(TEMPC) = *
*******************************    * 1H(+) $                     *
 *                                 *DASHES(LEFT'COL + 1, BOX'TAIL*
 *                                 * - 1, LINE'NO, 1) $          *
 *                                 *******************************
 *
****************************
*REAL'START = LEFT'COL +   *
* BOX'TAIL + 1 $           *
*BYTE($LEFT'COL + BOX'TAIL,*
* 1$)(TEMPC) = 1H(*) $     *
****************************
```

```
****************
* 52  FROM 48  *
****************
 *
 *
 *
******************************
*BYTE($LEFT'COL, 1$)(TEMPC) =*
* 1H(*) $                    *
*REAL'START = LEFT'COL + 1 $ *
******************************
```

DESIGN DIAGRAM OF CONTINUE'BOX

```
*********************************
*PROC CONTINUE'BOX(CONT'REC) $  *
*********************************
 *
 *
 *
 * **THE PROCEDURE CONTINUE'BOX
 *   APPENDS COMPATIBLE STATEMENT
 *   UNITS TO THE CURRENT FILE 3
 *   RECORD. A COMPATIBLE
 *   STATEMENT UNIT IS ONE WHICH
 *   CAN APPEAR IN THE SAME CODE
 *   BLOCK WITH THE STATEMENT
 *   UNITS WHICH HAVE ALREADY BEEN
 *   PLACED THERE. **
 *
 *
*********************************
*ITEM CONT'REC INTEGER $        *
*** REC NO. OF CONTINUED BLOCK. *
* **                            *
*********************************
 *
 *
 *
 * **CONTINUE A PREVIOUSLY
 *   INITIATED BOX **
 *
 *
****************
*IF DEBUG10 $  *   ********************************
***7**         *---*MESSAGE = 5H(CONT ) $         *
*              *   *T'MESS = IFORMAT(CONT'REC) $  *
****************   *MESSAGE = CAT(MESSAGE, T'MESS)*
 *                 * $                            *
 *                 *OUT(MESSAGE, MESS'SW) $       *
 *                 ********************************
 *
*********************************
*F4'END($ACCESS3(CONT'REC)$) =  *
* LAST'4 $                      *
*LINES($ACCESS3(CONT'REC)$) =   *
* LINES($ACCESS3(CONT'REC)$) +  *
* LINES'OUT($ACCESS4(FIRST'4)$) *
* $                             *
*BLOCK'WIDTH($ACCESS3(CONT'REC) *
*$) = MAX(BLOCK'WIDTH($ACCESS3( *
*CONT'REC)$), MAX'LINE'LNGTH($A *
*CCESS4(FIRST'4)$)) $           *
*WFITE3 = TRUE $                *
*RETURN $                       *
*********************************
```

```
****************************
*PROC CREATE'FILE4'RECS $  *
****************************
 *
 *
 *
 * '' THE PROCEDURE
 *   CREATE'FILE4'RECS CREATES A
 *   SET OF FILE 4 RECS FOR A
 *   STATEMENT UNIT IF IT IS
 *   APPROPRATE TO DO SO THE
 *   PROMARY FUNCTION OF
 *   CREATE'FILE4'RECS IS TO BREAK
 *   UP STATEMENT UNITS (IN FILE
 *   2) SO THAT THEY MAY BE
 *   DISPLAYED IN ACCORDANCE WITH
 *   THE VALUE OF ST'MAX. THE
 *   PROCEDURE ATTEMPTS TO BREAK
 *   UP STATEMENT UNITS AT THE
 *   SPACE CLOSEST TO ST'MAX. IF
 *   THERE IS NO BLANK, THEN THE
 *   BREAK IS MADE AT ST'MAX. A
 *   SET OF FILE 4 RECORDS
 *   CONSISTS OF A HEADER RECORD
 *   AND ANY NUMBER (I9INCLUDING 0
 *   )LINE BREAK RECORDS (SEE JSDD
 *   PROGRAM STRUCTURE SECTION
 *   4.5.1). ''
 *
 *
**************************************
*ITEM BYTES INTEGER $                *
*'' NO. OF BYTES PROCCESSED''        *
*ITEM CUR'PCS INTEGER $              *
*'' CURRENT POSITION IN LINE ''      *
*                                    *
*ITEM DONE B $                       *
*'' TERMINATION FLAG ''              *
*ITEM F1'REC INTEGER $               *
*'' THE CURRENT FILE 1 RECORD''      *
*                                    *
*ITEM LAST'BYTE INTEGER $            *
*'' BYTE POS OF LAST BREAK ''        *
*ITEM LAST'REC INTEGER $             *
*'' REC OF LAST BREAK ''             *
*ITEM STMT'TYPE INTEGER $            *
*'' PART 2 STATEMENT TYPE OF         *
* THE REC''                          *
*ITEM TEMP'BYTE INTEGER $            *
*''USED TO STORE THE BREAK           *
* BYTE''                             *
*ITEM TEMP'REC INTEGER $             *
*'' USED TO STORE THE BREAK          *
* RECORD''                           *
*BYTES = 1 $                         *
*LAST'BYTE = 1 $                     *
```

```
C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER
DESIGN DIAGRAM OF CREATE'FILE4'RECS

*********************************
 *
 *
***************
*IF DEBUG11 $ *    ********************************
*''7''        *---*MESSAGE = 9H(CREATE 4 ) $      *
*             *    *T'MESS = IFORMAT(LAST'4 + 1) $*
***************    *                              *
 *                 *MESSAGE = CAT(MESSAGE, T'MESS)*
 *                 * $                            *
 *                 *OUT(MESSAGE, MESS'SH) $       *
 *                 ********************************
 *
*********************************
*FILE3'INCLUSION = TRUE $       *
*'' ASSUME IT WILL BE A FILE3   *
* ENTRY ''                      *
*STMT'TYPE = BOX'MAP(STMT'TOKEN*
*) $                            *
*********************************
 *
 *
*********    *********************************
*IFEITH  *-+-*STMT'TYPE EQ BLOCK'DELIM OR    *
*********  * * STMT'TYPE EQ DELIM'COMMENT $  *----+
 *         * *                               *    * ''DON'T PUT IN FILE 3 ''
 *         * *********************************    *
 *         *                                      *
 *         *                                      ****************************
 *         *                                      *FILE3'INCLUSION = FALSE $ *
 *         *                                      *RETURN $                  *
 *         *                                      ****************************
 *         *
 *         *  *********************************   ***********
 *         +-*ORIF STMT'TYPE EQ END'SCOPE $  *---*RETURN $ *
 *            *********************************   ***********
 *
 *
 * '' STMT'TYPE IS A PRINTING
 *  STMT UNIT. IF ITS LENGTH ''
 * '' EXCEEDS ST'MAX, THEN BREAK
 *  IT UP. ''
 *
 *
*******************************
*FIRST'4 = LAST'4 + 1 $       *
*'' NEXT FILE 4 REC. ''       *
*LAST'4 = FIRST'4 $           *
*LAST'REC = FILE2'INDEX $     *
*F2'PTR($ACCESS4(FIRST'4)$) =*
* LAST'REC $                  *
*WRITE4 = TRUE $              *
*******************************
 *
 *
 *
 *
```

```
DESIGN DIAGRAM OF CREATE'FILE4'RECS

*********    **************************
*IFEITH  *-+-*STMT'LNGTH LQ ST'MAX $  *----+
*********  ' **************************    ' '' STMT UNIT WILL FIT ON 1
 '         '                               '   LINE ''
 '         '                               '
 '         '                               '
 '         '                              ********************************
 '         '                              *LINES'OUT($ACCESS4(FIRST'4)$) *
 '         '                              * = 1 $                        *
 '         '                              *MAX'LINE'LNGTH($ACCESS4(FIRST'*
 '         '                              *4)$) = STMT'LNGTH $           *
 '         '                              ********************************
 '         '
 '         ' ***********
 '         +-'ORIF 1 $  '----+
 '           ***********     ' '' STMT UNIT MUST SPAN LINES
 '                           '  -- TRY TO BREAK IT AT A
 '                           '  SPACE.''
 '                           ''' IF NOT POSSIBLE, BREAK IT
 '                           '  AT ST'MAX ''
 '                           '
 '                           '
 '                          ********************************
 '                          *MAX'LINE'LNGTH($ACCESS4(FIRST'*
 '                          *4)$) = 0 $                    *
 '                          *DONE = FALSE $                *
 '                          ********************************
 '                           '
 '                           '
 '                          **********************    *********************************
 '                          *DO WHILE [NOT DONE]  *---*LAST'4 = LAST'4 + 1 $        *
 '                          **********************    *TEMP'BYTE = LAST'BYTE + ST'MAX*
 '                           '                        * $                             *
 '                           '                        *CUR'POS = BYTES + ST'MAX $     *
 '                           '                        *********************************
 '                           '                         '
 '                           '                         '
 '                           '                         '
 '                           '                         ' '' DOES THIS INCREMENT PUT US
 '                           '                         '   IN NEXT F2REC ''
 '                           '                         '
 '                           '                         '
 '                           '                        *********    **************************
 '                           '                        *IFEITH  *-+-*TEMP'BYTE GR MAXCOL $  *----+
 '                           '                        *********  ' **************************    ' '' YES -- IT DOES ''
 '                           '                         '         '                               '
 '                           '                         '         '                               '
 '                           '                         '         '                              ********************************
 '                           '                         '         '                              *TEMP'REC = LAST'REC + 1 $     *
 '                           '                         '         '                              *TEMP'BYTE = TEMP'BYTE - MAXCOL*
 '                           '                         '         '                              * $                            *
 '                           '                         '         '                              ********************************
 '                           '                         '         '
 '                           '                         '         ' ***********
 '                           '                         '         +-'ORIF 1 $  '----+
                                                                   ***********     ' '' NO -- STILL IN SAME REC ''
 .                           .                         .                           .
 .                           .                         .                           ************************
 .                           .                         .                           *TEMP'REC = LAST'REC $ *
 .                           .                         .                           ************************
 .                           .                         .
 .                           .                         .
 .                           .                         . '' SCAN BACK IN THE LINE FOR
 .                           .                         .  AN ACCEPTABLE BREAKPOINT ''
 .                           .                         .
 .                           .                        *********************************
 .                           .                        *DO WHILE [CUR'POS - BYTES GQ   *
 .                           .                        * LOW'LIM AND SUBSTR(F2'LINE($A *      **************************
 .                           .                        *CCESS2(TEMP'REC)$), TEMP'BYTE, * *---*CUR'POS = CUR'POS - 1 $ *
 .                           .                        * 1) NQ SPACES(1)]              *      **************************
 .                           .                        *                               *       .
 .                           .                        *********************************       .
 .                           .                         .                                      .
 .                           .                         .                                     ******
 .                           .                         .                                     * 59 *
 .                           .                         .                                     ******
 .                           .                         .
 .                           .                         . '' HAVE WE FOUND AN ACCEPTABLE
 .                           .                         .  BREAKPOINT  ''
 .                           .                         .
 .                           .                        *********************************
 .                           .                        *IF CUR'POS - BYTES LS LOW'LIM  *
 .                           .                        * $                             *----*
 .                           .                        *                               *    . '' NO -- 3REAK IT AT ST'MAX ''
 .                           .                        *********************************    .
 .                           .                         .                                   .
 .                           .                         .                                   ********************************
 .                           .                         .                                   *CUR'POS = BYTES + ST'MAX $    *
 .                           .                         .                                   *TEMP'BYTE = LAST'BYTE + ST'MAX*
 .                           .                         .                                   * $                            *
 .                           .                         .                                   ********************************
 .                           .                         .                                   .
 .                           .                         .                                   .
 .                           .                         .                                   ******
 .                           .                         .                                   * 60 *
 .                           .                         .                                   ******
 .                           .                         .
 .                           .                         . '' PUT INFO IN FILE 4 ''
 .                           .                         .
 .                           .                        *******************************
 .                           .                        *F2'BYTE($ACCESS4(LAST'4)$) = *
 .                           .                        * TEMP'BYTE $                 *
 .                           .                        *F2'REC($ACCESS4(LAST'4)$) =  *
 .                           .                        * TEMP'REC $                  *
 .                           .                        *F2'PTR($ACCESS4(LAST'4)$) = 0*
```

```
  .                           .                        * $                            *
  .                           .                        *WRITE4 = TRUE $               *
  .                           .                        ********************************
  .                           .                         .
  .                           .                         .
  .                           .                         . '' JPDATE MAX'LINE'LNGTH ''
  .                           .                         .
  .                           .                        ********************************
  .                           .                        *MAX'LINE'LNGTH($ACCESS4(FIRST'*
  .                           .                        *4)$) = MAX(MAX'LINE'LNGTH($ACC*
  .                           .                        *ESS4(FIRST'4)$), CUR'POS -    *
  .                           .                        * BYTES) $                     *
  .                           .                        *WRITE4 = TRUE $               *
  .                           .                        ********************************
  .                           .                         .
  .                           .                         .
  .                           .                         . '' ANY MORE BREAKS REQUIRED
  .                           .                         .   ''
  .                           .                         .
  .                           .                         .
  .                           .                        ******
  .                           .                        * 61 *
  .                           .                        ******
  .                           .
  .                          ***************
  .                          *IF DEBUG11 $  *    ********************************
  .                          *''7''         *---*MESSAGE = 11H(END CREATE ) $ *
  .                          *              *    *T'MESS = IFORMAT(LAST'4) $    *
  .                          ***************     *MESSAGE = CAT(MESSAGE, T'MESS)*
  .                                              * $                            *
  .                                              *OUT(MESSAGE, MESS'SW) $       *
  .                                              ********************************
**********
*RETURN $ *
**********
```

```
DESIGN DIAGRAM OF CREATE"FILE4"RECS

****************
* 59  FROM 57  *
****************
  .
  .
  .
*********    *******************
*IFEITH  *-+-*TEMP"BYTE EQ 1 $  *----+
*********  . *******************     .  ""MOVE BACK TO PREVIOUS REC ""
           .                         .
           .                         .
           .                         .
           .                        ****************************
           .                        *TEMP"REC = TEMP"REC - 1 $ *
           .                        *TEMP"BYTE = MAXCOL $      *
           .                        ****************************
           .
           . ***********     ********************************
           +-*ORIF 1 $  *---*TEMP"BYTE = TEMP"BYTE - 1 $ *
             ***********     ********************************
```

```
DESIGN DIAGRAM OF CREATE'FILE4'RECS

****************
* 60  FROM 57  *
****************
 *
 *
 *
***************************
*IF TEMP'BYTE GR MAXCOL $  *-----+
***************************      * ** MOVE TO NEXT RECORD **
                                 *
                                 *
                               *********************************
                               *TEMP'BYTE = TEMP'BYTE - MAXCOL*
                               * $                             *
                               *TEMP'REC = TEMP'REC + 1 $      *
                               *********************************
```

```
C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER                                                       PAGE 61
DESIGN DIAGRAM OF CREATE'FILE4'RECS

****************
* 61  FROM 58  *
****************
 *
 *
 *
**********   ********************************
*IFEITH  *-+-*CUR'POS + ST'MAX GQ STMT'LNGTH *
**********  * * $                              *----+
 *          * *                               *     * ** NO WE'RE THROUGH **
 *          * ********************************      *
 *          *                                       *
 *          *                                      ********************************
 *          *                                      *LINES'OUT($ACCESS4(FIRST'4)$) *
 *          *                                      * = LAST'4 - FIRST'4 + 1 $     *
 *          *                                      *MAX'LINE'LNGTH($ACCESS4(FIRST'*
 *          *                                      *4)$) = MAX(MAX'LINE'LNGTH($ACC*
 *          *                                      *ESS4(FIRST'4)$), STMT'LNGTH - *
 *          *                                      * CUR'POS) $                   *
 *          *                                      *DONE = TRUE $                 *
 *          *                                      ********************************
 *          *
 *          * ***********
 *          +-*ORIF 1 $  *----+
 *            ***********     * ** AT LEAST 1 MORE BREAK
 *                            *  NEEDED **
 *                            *
 *                            *
 *                           **************************
 *                           *LAST'BYTE = TEMP'BYTE $ *
 *                           *LAST'REC = TEMP'REC $   *
 *                           *BYTES = CUR'POS $       *
 *                           **************************
 *
******************
*WRITE4 = TRUE $ *
******************
```

```
DESIGN DIAGRAM OF CREATE*H*PTR*REC

********************************
*PROC CREATE*H*PTR*REC(FATHER = *
* SON) $                          *
*                                *
********************************
 *
 *
 *
 * ** THE PROCEDURE
 *   CREATE*H*PTR*REC SETS UP
 *   POINTERS WHICH HAVE THE
 *   EFFECT OF INSERTING AN H*SON
 *   INTO THE FILE 3 TREE**
 *
 *
*****************************
*ITEM FATHER INTEGER $      *
*** THE FATHER OF THE NEW   *
* RECORD **                 *
*ITEM SON INTEGER $         *
*** THE RECORD NUMBER OF THE*
* NEW RECORD**              *
*****************************
 *
 *
***************
*IF DEBUG12 $ *    *******************************
***7**        *---*MESSAGE = 9H(CREATE H ) $     *
*             *    *T*MESS = IFORMAT(F3*AVAIL) $ *
***************    *MESSAGE = CAT(MESSAGE, T*MESS)*
 *                 * $                           *
 *                 *OUT(MESSAGE, MESS*SW) $      *
 *                 *******************************
 *
********************************
*SCN = F3*AVAIL $              *
*F3*AVAIL = F3*AVAIL + 1 $     *
*H*PTR($ACCESS3(FATHER)$) = SON*
* $                            *
*WRITE3 = TRUE $               *
*BACK*H($ACCESS3(SON)$) =      *
* FATHER $                     *
*BACK*V($ACCESS3(SON)$) = 0 $  *
*H*PTR($ACCESS3(SON)$) = 0 $   *
*V*PTR($ACCESS3(SON)$) = 0 $   *
*WRITE3 = TRUE $               *
*RETURN $                      *
********************************
```

DESIGN DIAGRAM OF CREATE'V'PTR'REC

```
*********************************
*PROC CREATE'V'PTR'REC(FATHER = *
* SON) $                          *
*                                 *
*********************************
 *
 *
 *
 * ** THE PROCEDURE CREATE
 *   V'PTR'REC IS THE SAME AS
 *   CREATE'H'PTR'REC EXCEPT THAT
 *   IT CREATES A V'SON. **
 *
 *
*********************************
*ITEM FATHER INTEGER $          *
***RECORD NUMBER OF THE NEW     *
* REC'S FATHER**                *
*ITEM SON INTEGER $             *
*** RECORD NUMBER OF NEW FILE 3*
* REC **                        *
*********************************
 *
 *
***************
*IF DEBUG12 $ *        *********************************
***7**          *---*MESSAGE = 9H(CREATE V ) $      *
*             *        *T'MESS = IFORMAT(F3'AVAIL) $  *
***************        *MESSAGE = CAT(MESSAGE, T'MESS)*
 *                     * $                             *
 *                     *OUT(MESSAGE, MESS'SH) $        *
 *                     *********************************
 *
*********************************
*SON = F3'AVAIL $               *
*F3'AVAIL = F3'AVAIL + 1 $      *
*V'PTR($ACCESS3(FATHER)$) = SON*
* $                             *
*WRITE3 = TRUE $                *
*BACK'V($ACCESS3(SON)$) =       *
* FATHER $                      *
*BACK'H($ACCESS3(SON)$) = 0 $   *
*H'PTR($ACCESS3(SON)$) = 0 $    *
*V'PTR($ACCESS3(SON)$) = 0 $    *
*WRITE3 = TRUE $                *
*RETURN $                       *
*********************************
```

DESIGN DIAGRAM OF DASHES

```
********************************
*PROC DASHES(COL, LNGTH, LINE, *
* OUTPT) $                       *
*                              *
********************************
 *
 *
 *
 * '' THE PROCEDURE DASHES
 *  OUTPUTS A STRING OF DASHES OF
 *  LENGTH LNGTH, STARTING ON
 *  COLUMN COL OF PUTOUT RECORD
 *  LINE OR TEMPC (DEPENDING ON
 *  THE VALUE OF OUTPT). ''
 *
 *
********************************
*ITEM COL INTEGER $            *
*'' THE COLUMN ON WHICH DASHES *
* START ''                     *
*ITEM LINE INTEGER $           *
*''PUTOUT RECORD ON WHICH      *
* DASHES APPEAR''              *
*ITEM LNGTH INTEGER $          *
*'' LENGTH OF DASH STRING TO BE*
* OUTPUT''                     *
*ITEM DASH H 132 P             *
* 132H(------------------------*
*------------------------------*
*------------------------------*
*------------------------------*
*--          ---------------) $ *
*ITEM OUTPT B $                *
*'' OUTPUT TO TEMP OR OUT'LINE *
* ''                           *
********************************
 *
 *
*********    **********    **$r*************************
*IFEITH  *-+-*OUTPT $  *---*BYTE($COL, LNGTH$)(TEMPC) =*
*********  * **********    * BYTE($0, LNGTH$)(DASH) $  *
 *         *               *****************************
 *         *
 *         * ***********     *********************************
 *         +-*ORIF 1 $  *---*BYTE($COL, LNGTH$)(OUT'LINE($A*
 *           ***********     *CCESS'OUT(LINE)$)) = BYTE($0, *
 *                           * LNGTH$)(DASH) $               *
 *                           *********************************
 *
***********
*RETURN $ *
***********
```

```
********************************
*PROC DISPLAY'STUMP'REF(HORIZ) *
* $                            *
*                              *
********************************
 *
 *
 *
 *  '' THE PROCEDURE
 *   DISPLAY'STUMP'REF CREATES AND
 *   OUTPUTS A STUMP REFERENCE
 *   DISPLAY BOX EITHER
 *   HORIZONTALLY OR VERTICALLY
 *   FROM THE CURRENT DIAGRAM CODE
 *   BLOCK. THE BOX CONTAINS THE
 *   REFERENCE NUMBER OF THE STUMP
 *   WHICH IS OBTAINED FROM THE
 *   PAGE'REF ITEM OF THE STUMP'S
 *   GROUP ENTRY. ''
 *
 *
********************************
*ITEM BOTTOM INTEGER $         *
*'' THE BOTTOM LINE OF THE     *
* DISPLAY BOX''                *
*ITEM DISPLAY INTEGER $        *
*''LINE ON WHICH  REFERENCE    *
* APPEARS''                    *
*ITEM HORIZ B $                *
*''TYPE OF DISPLAY. HORIZ OR   *
* VERT.''                      *
*ITEM INDEX INTEGER $          *
*'' THE INDEX OF THE STUMP'S   *
* GROUP ENTRY''                *
*ITEM LEFT'START INTEGER $     *
*''THE STARTING COLUMN OF THE  *
* BOX''                        *
*ITEM LNGTH INTEGER $          *
*'' USED TO STORE STRING       *
* LENGTHS''                    *
*ITEM LINE INTEGER $           *
*'' DISPLACEMENT OF LINES ON   *
* CURRENT PAGE''               *
*ITEM PAGE INTEGER $           *
*'' DUMMY VARIABLE FOR CALL TO *
* PEMQUO ''                    *
*ITEM TOP INTEGER $            *
*''THE TOP LINE OF BOX''       *
********************************
 *
 *
****************
*IF DEBUG13 $  *    ********************************
*''7''         *----*MESSAGE = 11H(DISP STUMP ) $  *
*              *    *T'MESS = IFORMAT(CUR'REC) $   *
****************    *MESSAGE = CAT(MESSAGE, T'MESS)*
```

DESIGN DIAGRAM OF DISPLAY'STUMP'REF

```
                    * $                             *
 .                  *OUT(MESSAGE, MESS'SM) $        *
 .                  *********************************
 .
*********   **********     ****************************
*IFEITH  *-+-*HORIZ $  *---*DASHES(RIGHT'COL + 1,     *
*********  . **********    * H'SPACE'2, MIDPOINT, 0) $*
 .         .               *DISPLAY = MIDPOINT $      *
 .         .               *TOP = MIDPOINT - 1 $      *
 .         .               *BOTTOM = MIDPOINT + 1 $   *
 .         .               *LEFT'START = RIGHT'COL +  *
 .         .               * H'SPACE'2 + 1 $          *
 .         .               ****************************
 .         .
 .         . ***********   ********************************
 .         +-*ORIF 1 $  *---*TOP = BOTTOM'LINE + V'SPACE +*
 .           ***********   * 1 $                          *
 .                         *BOTTOM = TOP + 2 $            *
 .                         *DRAW'LINE(LEFT'COL + 1,       *
 .                         * BOTTOM'LINE, LEFT'COL + 1,   *
 .                         * TOP) $                       *
 .                         *DISPLAY = TOP + 1 $           *
 .                         *LEFT'START = LEFT'COL $       *
 .                         ********************************
 .
***************     ******************************
*IF HEADING $  *---*REMQUO(TOP, PAGE'LNGTH = PAGE,*
***************     * LINE) $                     *
 .                  ******************************
 .                  .
 .                  .
 .                  ***********************     **********   ***********   ********************************
 .                  *IF LINE LS HEADROOM $ *---*IFEITH   *-+-*HORIZ $  *---*TOP = PAGE * PAGE'LNGTH - 1 $ *
 .                  ***********************     ********** . ***********   ********************************
 .                  .                                      .
 .                  .                                      .
 .                  .                                      . ************   ***************************
 .                  .                                      +-*ORIF 1 $  *---*TOP = PAGE * PAGE'LNGTH +*
 .                  .                                        ************   * HEADROOM $              *
 .                  .                                                       ***************************
 .                  .
 .                  ****************************
 .                  *REMQUO(BOTTOM, PAGE'LNGTH =*
 .                  * PAGE, LINE) $             *
 .                  ****************************
 .                  .
 .                  .
 .                  ***********************   ******************************
 .                  *IF LINE LS HEADROOM $ *---*BOTTOM = PAGE * PAGE'LNGTH +*
 .                  ***********************   * HEADROOM + 1 $              *
 .                  .                         ******************************
 .                  .
 .                  ******************     *****************************
 .                  *IF NOT HORIZ $  *---*REMQUO(DISPLAY, PAGE'LNGTH =*
 .                  ******************     * PAGE, LINE) $              *
 .                                         *****************************
 .
```

```
DESIGN DIAGRAM OF DISPLAY'STUMP'REF

 *
 *                                      ************************   *******************************
 *                                      *IF LINE LS HEADROOM $ *---*DISPLAY = PAGE * PAGE'LNGTH +*
 *                                      ************************   * HEADROOM $                  *
 *                                                                 *******************************
 *
**********************
*INDEX = CUR'GROUP $ *
**********************
 *
 *
 *
 * ** FIND ELEMENT OF GROUP
 *  REFERRING TO THIS STUMP **
 *
 *
*********    **********    ********************************
*IFEITH *-+-*HORIZ $  *---*DO WHILE [(/ H'SON /) NQ     *   **************************
********* * **********     * F3'REC($INDEX$) AND INDEX LS *---*INDEX = NEXT($INDEX$) $ *
 *        *                * GROUP'AVAIL] DO WHILE [(/ H' *   **************************
 *        *                ********************************
 *        *
 *        * ************   ********************************
 *        +-*ORIF 1 $  *---*DO WHILE [(/ V'SON /) NQ     *   **************************
 *          ************   * F3'REC($INDEX$) AND INDEX LS *---*INDEX = NEXT($INDEX$) $ *
 *                         * GROUP'AVAIL]                 *   **************************
 *                         ********************************
 *
*****************************   **************************
*IF INDEX GQ GROUP'AVAIL $  *---*PH2ERR(19H(GROUP ERROR -*
*****************************   * STUMP)) $              *
 *                              **************************
 *
*********************************
*STARS(LEFT'START, 6, TOP) $    *
*BYTE($LEFT'START,              *
* 6$)(OUT'LINE($ACCESS'OUT(DISP*
*LAY)$)) = 6H(*    *) $         *
*TEMPC = IFORMAT(PAGE'REF($INDE*
*X$)) $                         *
*LNGTH = LENGTH(TEMPC) $        *
*BYTE($LEFT'START + 2,          *
* LNGTH$)(OUT'LINE($ACCESS'OUT(*
*DISPLAY)$)) = BYTE($6,         *
* LNGTH$)(TEMPC) $              *
*STARS(LEFT'START, 6, BOTTOM) $*
*                               *
*********************************
 *
 *
****************
*IF HEADING $  *----+
****************    * **UPDATE GROUP STORE WITH
 *                  * FROM'PAGE **
 *                  *
 *                  *
 *                 *******************************
```

```
                    *FROM'PAGE($INDEX$) = PAGE'NO $*
  .                 *                               *
  .                 *********************************
  .
***********
*RETURN $ *
***********
```

DESIGN DIAGRAM OF DOTS

```
**********************************
*PROC DOTS(COL, LNGTH, LINE) $   *
**********************************
 *
 *
 *
 * ** THE PROCEDURE DOTS OUTPUTS
 *   A STRING OF  LNGTH DOTS
 *   STARTING AT COLUMN COL OF THE
 *   LINE TH RECORD OF PUTOUT. **
 *
 *
**********************************
*ITEM COL INTEGER $              *
*** THE STARTING COLUMN **       *
*ITEM LINE INTEGER $             *
*** THE NUMBER OF THE PUTOUT     *
* RECORD**                       *
*ITEM LNGTH INTEGER $            *
*** NUMBER OF DOTS TO BE OUTPUT*
* **                             *
*ITEM DOT H 132 P                *
* 132H(..........................*
*................................*
*................................*
*................................*
*..         .................) $ *
*BYTE($COL, LNGTH$)(OUT*LINE($A*
*CCESS*OUT(LINE)$)) = BYTE($0, *
* LNGTH$)(DOT) $                 *
*RETURN $                        *
**********************************
```

```
*********************************
*PROC DRAW*LINE(FROM*X, FROM*Y, *
* TO*X, TO*Y) $                 *
*                               *
*********************************
 *
 *
 *
 * ** THE PROCEDURE DRAW*LINE
 *   DRAWS A LINE FROM THE POINT
 *   HAVING X AND Y COORDINATES
 *   (IN TERMS OF COLUMN AND
 *   RECORD) FROM*X, FROM*Y TO THE
 *   POINT TO*X, TO*Y. **
 *
 *
*****************************
*ITEM FROM*X INTEGER $      *
***STARTING X COORD**       *
*ITEM FROM*Y INTEGER $      *
*** STARTING Y COORD **     *
*ITEM LINE INTEGER $        *
*** USED TO STORE PAGE      *
* DISPLACEMENT **           *
*ITEM PAGE INTEGER $        *
***DUMMY USED IN CALL TO    *
* REMQUO**                  *
*ITEM TO*X INTEGER $        *
*** ENDING  X COORD **      *
*ITEM TO*Y INTEGER $        *
*** ENDING Y COORD **       *
*ITEM TWO*LINES B $         *
***ON WHEN 2 LINES NEEDED TO*
* CONNECT POINTS**          *
*TWO*LINES = FALSE $        *
*****************************
 *
 *
***************
*IF DEBUG14 $ *    *******************************
***18**           *---*TEMPC = 1H(*) $               *
*             *       *MESSAGE = 10H(START DRAW) $   *
***************       *OUT(MESSAGE, MESS*SW) $       *
 *                    *MESSAGE = IFORMAT(FROM*X) $   *
 *                    *MESSAGE = CAT(TEMPC, MESSAGE)*
 *                    * $                            *
 *                    *OUT(MESSAGE, MESS*SW) $       *
 *                    *MESSAGE = IFORMAT(FROM*Y) $   *
 *                    *MESSAGE = CAT(TEMPC, MESSAGE)*
 *                    * $                            *
 *                    *OUT(MESSAGE, MESS*SW) $       *
 *                    *MESSAGE = IFORMAT(TO*X) $     *
 *                    *MESSAGE = CAT(TEMPC, MESSAGE)*
 *                    * $                            *
 *                    *OUT(MESSAGE, MESS*SW) $       *
 *                    *MESSAGE = IFORMAT(TO*Y) $     *
```

```
DESIGN DIAGRAM OF DRAW°LINE
                    *MESSAGE = CAT(TEMPC, MESSAGE)*
 *                  * $                           *
 *                  *OUT(MESSAGE, MESS°SW) $      *
 *                  *******************************
 *
*******************************
*IF FROM°X NQ TO°X AND FROM°Y *
* NQ TO°Y $                   *----+
*                             *    * **NEED A PAIR OF PERPENDICULAR
*******************************    *  LINES **
 *                                 *
 *                                 *
 *                                 ********************************
 *                                 *BYTE($TO°X, 1$)(OUT°LINE($ACCE*
 *                                 *SS°OUT(FROM°Y)$)) = 1H(+) $   *
 *                                 *TWO°LINES = TRUE $            *
 *                                 ********************************
 *
*******************************
*IF (FROM°Y EQ TO°Y OR        *
* TWO°LINES) AND TO°X - FROM°X *----+
* GR 1 $                      *    * ** A HORIZONTAL **
*******************************    *
 *                                 *
 *                                 ***************************
 *                                 *DASHES(FROM°X + 1, TO°X -*
 *                                 * FROM°X - 1, FROM°Y, 0) $*
 *                                 ***************************
 *
*******************************
*IF (FROM°X EQ TO°X OR        *
* TWO°LINES) AND TO°Y - FROM°Y *----+
* GR 1 $                      *    * ** A VERTICAL **
*******************************    *
 *                                 *
 *                                 ******************************
 *                                 *FOR I = FROM°Y + 1, 1, TO°Y - *   **********   ************   ******
 *                                 * 1 $                         *---*IFEITH  *-+-*HEADING $ *--* 72 *
 *                                 *                             *   ********** * ************   ******
 *                                 ******************************               *
 *                                                                              *
 *                                                                              * ***********   ******
 *                                                                              +-*ORIF 1 $ *--* 73 *
 *                                                                                ***********   ******
 *
***************
*IF DEBUG14 $ *   ***************************
*°°5°°        *---*MESSAGE = 8H(END DRAW) $ *
*             *   *OUT(MESSAGE, MESS°SW) $  *
***************   ***************************
 *
 *
***********
*RETURN $ *
***********
```

```
DESIGN DIAGRAM OF DRAW'LINE

****************
* 72  FROM 71  *
****************
 *
 *
 *
******************************
*REMQUO(I, PAGE'LNGTH = PAGE,*
* LINE) $                    *
******************************
 *
 *
**********   ***************   *********************
*IFEITH  *-+-*LINE EQ 0 $  *---*OUTPUT'HEADER(I) $ *
********** * ***************   *********************
           *
           *
           * ***************************   *********************************
           +-*ORIF LINE GR HEADROOM $  *---*BYTE($TO'X, 1$)(OUT'LINE($ACCE*
             ***************************   *SS'OUT(I)$)) = 1H(') $         *
                                           *********************************
```

```
DESIGN DIAGRAM OF DRAW"LINE

*****************
* 73  FROM 71   *
*****************
 *
 *
 *
*********************************
*BYTE($TO"X, 1$)(OUT"LINE($ACCE*
*SS"OUT(I)$)) = 1H(") $         *
*********************************
```

```
DESIGN DIAGRAM OF EXTRACT'NAME

**************************
*PROC EXTRACT'NAME() $  *
**************************
 *
 *
 *
 * ** THE FUNCTION EXTRACT'NAME
 *   EXTRACTS AND RETURNS THE NAME
 *   OF A MODULE  HEAD. **
 *
 *
*************************************
*ITEM DONE B $                      *
*** TERMINATION FLAG **             *
*ITEM EXTRACT'NAME CHARACTER $      *
*** RETURNS THE MODULE NAME**       *
*ITEM INDEX INTEGER $               *
***INDEX INTO TEXT STRING **        *
*ITEM INDEX1 INTEGER $              *
*** INDEX INTO TEXT STRING**        *
*ITEM INDEX'F2 INTEGER $            *
*** FILE 2 RECORD NUMBER**          *
*ITEM LNGTH INTEGER $               *
*** USED TO STORE LENGTHS OF        *
* STRINGS**                         *
*ITEM SOURCE'LINE CHARACTER $       *
*** STORES FILE 2 RECORD TEXT***
*                                   *
*INDEX'F2 = 0 $                     *
*DONE = FALSE $                     *
*************************************
 *
 *
***********************************
*DO WHILE (INDEX'F2 LS F2'RECS *     *********************************
* AND NOT DONE)                *---*SOURCE'LINE = F2'LINE($ACCESS2*
*                              *    *(FILE2'INDEX + INDEX'F2)$) $  *
***********************************  *LNGTH = LENGTH(SOURCE'LINE) $ *
 *                                   *INDEX = 6 $                   *
 *                                   *********************************
 *                                    *
 *                                    *
 *                                   *********************************
 *                                   *DO WHILE (NOT DONE AND INDEX - *   ******
 *                                   * 1 LQ LNGTH)                   *--* 77 *
 *                                   *                               *   ******
 *                                   *********************************
 *                                    *
 *                                    *
 *                                   ******************************
 *                                   *INDEX'F2 = INDEX'F2 + 1 $ *
 *                                   ******************************
 *
**********  *************
*IFEITH *-+-*NOT DONE $ *----+
********** * *************    * **NO PROC OR CLOSE -- HAVE
```

```
DESIGN DIAGRAM OF EXTRACT'NAME

                                   .  MAIN'PGM HEAD **
 .      .                          .
 .      .                          .
 .      .                         ********************************
 .      .                         *EXTRACT'NAME = PGM'NAME $      *
 .      .                         *EXTRACT'NAME = CNVERT(EXTRACT'*
 .      .                         *NAME) $                        *
 .      .                         ********************************
 .      .
 .      . ***********
 .      +-*ORIF 1 $  *----+
 .        ***********     . **FIND PROC NAME **
 .                        .
 .                        .
 .                        *********************************
 .                        *DO WHILE (BYTE($INDEX,         * *********************
 .                        * 1$)(SOURCE'LINE) EQ 1H( ) AND *---*INDEX = INDEX + 1 $ *
 .                        * INDEX LQ LNGTH)               * *********************
 .                        *********************************   .
 .                        .                                   .
 .                        .                                   .
 .                        .                                   . ** ADVANCE BEYOND SPACES **
 .                        .                                   .
 .                        .
 .                        *************************    ***************************
 .                        *IF INDEX - 5 GR LNGTH $ *---*PH2ERR(18H(CONTROL HEAD*
 .                        *************************    * ERROR)) $                *
 .                        .                            ***************************
 .                        .
 .                        *****************************
 .                        *INDEX1 = INDEX $            *
 .                        *** SAVE STARTING POINT ** *
 .                        *****************************
 .                        .
 .                        .
 .                        *********************************
 .                        *DO WHILE (BYTE($INDEX,          *
 .                        * 1$)(SOURCE'LINE) NQ 1H( ) AND  *
 .                        * BYTE($INDEX, 1$)(SOURCE'LINE)   *     *********************
 .                        * NQ 1H(() AND BYTE($INDEX,         *---*INDEX = INDEX + 1 $ *
 .                        * 1$)(SOURCE'LINE) NQ 1H(DO WHI   *     *********************
 .                        * INDEX - 5 LQ LNGTH)            *       .
 .                        *                               *        .
 .                        *********************************        .
 .                        .                                        .
 .                        .                                        .
 .                        .                                        . ** NOTE THAT THIS WILL
 .                        .                                        .  TRUNCATE NAMES **
 .                        .                                        . ** SPANNING  FILE2 RECORDS. **
 .                        .                                        .
 .                        .                                        .
 .                        .
 .                        *********************************
 .                        *EXTRACT'NAME = SUBSTR(SOURCE'L*
 .                        *INE, INDEX1 - 5, INDEX -       *
 .                        * INDEX1) $                      *
 .                        *********************************
```

DESIGN DIAGRAM OF EXTRACT'NAME

```
****************
*IF DEBUG15 $  *    ***********************************
*""6""          *---*MESSAGE = 13H(EXTRACT NAME ) $*
*              *    *                                 *
****************    *MESSAGE = CAT(MESSAGE,           *
 *                  * EXTRACT'NAME) $                 *
 *                  *OUT(MESSAGE, MESS'SW) $          *
 *                  ***********************************
 *
************
*RETURN $ *
************
```

```
DESIGN DIAGRAM OF EXTRACT'NAME

****************
* 77  FROM 74  *
****************
 *
 *
 *
**************************
*IF INDEX - 1 LQ LNGTH $  *----+
**************************     * ''CHECK FOR AN OCCURRENCE OF
 *                             *  'PROC'. ''
 *                             *
 *                             *
 *                             *******************************
 *                             *IF BYTE($INDEX,               *
 *                             * 4$)(SOURCE'LINE) EQ 4H(PROC)  *----+
 *                             * $                             *    * '' HAVE FOUND 'PROC' . ''
 *                             *******************************     *
 *                                                                 *
 *                                                                 *********************
 *                                                                 *INDEX = INDEX + 4 $ *
 *                                                                 *DONE = TRUE $       *
 *                                                                 *********************
 *
*****************************
*IF NOT DONE AND INDEX - 2 LQ *
* LNGTH $                     *----+
*                             *    * '' CHECK FOR 'CLOSE'. ''
*****************************      *
 *                                 *
 *                                 ********************************
 *                                 *IF BYTE($INDEX,                *
 *                                 * 5$)(SOURCE'LINE) EQ 5H(CLOSE)  *----+
 *                                 * $                              *    * '' GOT IT ''
 *                                 ********************************     *
 *                                                                      *
 *                                                                      *********************
 *                                                                      *INDEX = INDEX + 5 $ *
 *                                                                      *DONE = TRUE $       *
 *                                                                      *********************
 *
****************    *********************
*IF NOT DONE $  *---*INDEX = INDEX + 1 $ *
****************    *********************
```

DESIGN DIAGRAM OF EXTRACT'TEXT

```
*******************************
*PROC EXTRACT'TEXT(F4'PTR) $  *
*******************************
 *
 *
 *
 * '' THE FUNCTION EXTRACT'TEXT
 *   EXTRACTS FILE2 TEXT IN
 *   ACCORDANCE WITH THE LINE
 *   BREAK INFORMATION CONTAINED
 *   IN THE SET OF FILE 4 RECORDS
 *   WHICH CORRESPOND TO THE
 *   STATEMENT UNIT. ''
 *
 *
*********************************
*ITEM BYTE'PTR INTEGER $        *
*''LAST BYTE OF TEXT TO BE      *
* RETURNED''                    *
*ITEM EXTRACT'TEXT CHARACTER $  *
*''CONTAINS TEXT TO BE          *
* RETURNED''                    *
*ITEM F4'PTR INTEGER $          *
*'' CURRENT FILE 4 RECORD ''    *
*ITEM FIELD1 INTEGER $          *
*'' STORES  F2'PTR OF CURRENT   *
* FILE4 REC''                   *
*ITEM FIELD2 INTEGER $          *
*''STORES LINES'OUT OR F2'REC   *
* OF REC''                      *
*ITEM FIELD3 INTEGER $          *
*''STORES MAX'LINE'LNGTH OR     *
* F2'BYTE OF REC''              *
*********************************
 *
 *
 *
 * '' LINE'PTR IS THE FILE 2
 *   RECORD WHICH CONTAINS''
 * ''THE LAST BYTE TO BE
 *   RETURNED''
 *
 *
*********************************
*ITEM LINE'PTR INTEGER $        *
*ITEM LNGTH INTEGER $           *
*''STORES STRING LENGTHS ''     *
*ITEM NEXT'F2 INTEGER $         *
*''NUMBER OF NEXT FILE 2 REC''  *
*ITEM NEXT'F4 INTEGER $         *
*''NUMBER OF NEXT FILE 4 REC''  *
*ITEM TEXT'LINE CHARACTER $     *
*''TEMPORARY STRING STORAGE''   *
*NEXT'F4 = F4'PTR + 1 $         *
*F4'PTR = ACCESS4(F4'PTR) $     *
*FIELD1 = F2'PTR($F4'PTR$) $    *
```

```
C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER                                                        PAGE 80
DESIGN DIAGRAM OF EXTRACT'TEXT

 .         .                                     .          . ***********     ********************************
 .         .                                     .          +-*ORIF 1 $   *---*EXTRACT'TEXT = SUBSTR(TEXT'LIN*
 .         .                                     .            ***********     *E, 1, F2'BYTE($NEXT'F4$) - 1) *
 .         .                                     .                            * $                            *
 .         .                                     .                            ********************************
 .         .                                     .
 .         .                                    ************************
 .         .                                    *FIRST'INVOC = FALSE $ *
 .         .                                    ************************
 .         .
 .         .
 .         . ***********      ********************************
 .         +-*ORIF 1 $   *----*TEXT'LINE = F2'LINE($ACCESS2(F*
 .           ***********      *IELD2)$) $                    *
 .                            ********************************
 .                             .
 .                             .
 .                            *********   ****************
 .                            *IFEITH  *-+-*NEXT'F4 GR 0 $  *----+
 .                            *********  . ****************      . '' NO EOFILE ''
 .                                       .                       .
 .                                       .                       .
 .                                       .                       .
 .                                       .                      ******
 .                                       .                      * 82 *
 .                                       .                      ******
 .                                       .
 .                                       . ***********
 .                                       +-*ORIF 1 $   *----+
 .                                         ***********      . ''THERE'S AN EOFILE ''
 .                                                          .
 .                                                          .
 .                                                         ********************************
 .                                                         *LNGTH = LENGTH(TEXT'LINE) $   *
 .                                                         *EXTRACT'TEXT = SUBSTR(TEXT'LIN*
 .                                                         *E, FIELD3, LNGTH - FIELD3 + 1)*
 .                                                         * $                            *
 .                                                         *NEXT'F2 = ACCESS2(FIELD2) $   *
 .                                                         ********************************
 .                                                          .
 .                                                          .
 .                                                         ********************
 .                                                         *IF NEXT'F2 GR 0 $  *----+
 .                                                         ********************     . '' THERE'S ANOTHER F2 REC FOR
 .                                                                                  .   THIS UNIT ''
 .                                                                                  .
 .                                                                                  .
 .                                                                                 ********************************
 .                                                                                 *TEXT'LINE = F2'LINE($NEXT'F2$)*
 .                                                                                 * $                            *
 .                                                                                 *EXTRACT'TEXT = CAT(EXTRACT'TEX*
 .                                                                                 *T, TEXT'LINE) $               *
 .                                                                                 ********************************
 .
 .
 .
 .
```

```
C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER                                                        PAGE 79
DESIGN DIAGRAM OF EXTRACT'TEXT

*FIELD2 = LINES'OUT($F4'PTR$) $*
*                              *
*FIELD3 = MAX'LINE'LNGTH($F4'PT*
*R$) $                         *
********************************
 .
 .
****************
*IF DEBUG16 $  *   ********************************
*''17''        *---*MESSAGE = 7H(EXTRACT) $       *
*              *   *T'MESS = IFORMAT(F4'PTR) $    *
****************   *MESSAGE = CAT(MESSAGE, T'MESS)*
 .                 * $                            *
 .                 *T'MESS = 1H(=) $              *
 .                 *OUT(MESSAGE, MESS'SW) $       *
 .                 *MESSAGE = IFORMAT(FIELD1) $   *
 .                 *MESSAGE = CAT(T'MESS, MESSAGE)*
 .                 * $                            *
 .                 *OUT(MESSAGE, MESS'SW) $       *
 .                 *MESSAGE = IFORMAT(FIELD2) $   *
 .                 *MESSAGE = CAT(T'MESS, MESSAGE)*
 .                 * $                            *
 .                 *OUT(MESSAGE, MESS'SW) $       *
 .                 *MESSAGE = IFORMAT(FIELD3) $   *
 .                 *MESSAGE = CAT(T'MESS, MESSAGE)*
 .                 * $                            *
 .                 *OUT(MESSAGE, MESS'SW) $       *
 .                 ********************************
 .
*******************************
*NEXT'F4 = ACCESS4(NEXT'F4) $ *
*******************************
 .
 .
*********    *******************************
*IFEITH  *-+-*FIELD1 NQ 0 OR FIRST'INVOC $ *----+
*********  . *******************************    . ** HAVE 1ST LINE OF STMT UNIT
 .         .                                    .  **
 .         .                                    .
 .         .                                    .
 .         .                                   ********************************
 .         .                                   *TEXT'LINE = F2'LINE($ACCESS2(F*
 .         .                                   *IELD1)$) $                    *
 .         .                                   ********************************
 .         .                                    .
 .         .                                    .
 .         .                                   *********    ****************
 .         .                                   *IFEITH  *-+-*FIELD2 EQ 1 $ *----+
 .         .                                   *********  . ****************    . ** ONLY 1 LINE **
 .         .                                    .         .                     .
 .         .                                    .         .                     .
 .         .                                    .         .                    *****************************
 .         .                                    .         .                    *EXTRACT'TEXT = TEXT'LINE $ *
 .         .                                    .         .                    *****************************
 .         .                                    .         .
 .         .                                    .         .
 .         .                                    .         .
```

DESIGN DIAGRAM OF EXTRACT'TEXT

```
****************
*IF DEBUG16 $  *     *******************************
*''2''          *---*OUT(EXTRACT'TEXT, MESS'SW) $ *
*              *     *******************************
****************
 .
 .
************
*RETURN $ *
************
```

```
****************
* 82  FROM 80  *
****************
 .
 .
 .
*********    ***************************
*IFEITH  *-+-*F2'PTR($NEXT'F4$) NQ 0 $ *-----+
*********  . ***************************     . ** NEXTF4 DOESN'T PERTAINS TO
           .                                 .   THE STMT UNIT **
           .                                 .
           .                                 .
           .                                ********************************
           .                                *LNGTH = LENGTH(TEXT'LINE) $   *
           .                                *EXTRACT'TEXT = SUBSTR(TEXT'LIN*
           .                                *E, FIELD3, LNGTH - FIELD3 + 1)*
           .                                * $                            *
           .                                ********************************
           .                                 .
           .                                 .
           .                                 .
           .                                 . ** SEE IF STMT UNIT CONTINUES
           .                                 .   TO NEXT F2 REC **
           .                                 .
           .                                 .
           .                                ********************************
           .                                *IF F2'PTR($NE XT'F4$) NQ FIELD2 *
           .                                * + 1 AND LNGT H EQ MAXCOL $   *------+
           .                                *                              *      . **IT DOES **
           .                                ********************************      .
           .                                                                      .
           .                                                                     ********************************
           .                                                                     *TEXT'LINE = F2'LINE($ACCESS2(F*
           .                                                                     *IELD2 + 1)$) $                *
           .                                                                     *EXTRACT'TEXT = CAT(EXTRACT'TEX*
           .                                                                     *T, TEXT'LINE) $               *
           .                                                                     ********************************
           .
           .
           . ***********
           +-*ORIF 1 $ *-----+
             ***********     . ** ANOTHER F4 REC FOLLOWS **
                             .
                             .
                            ********************************
                            *LINE'PTR = F2'REC($NEXT'F4$) $*
                            *                              *
                            *BYTE'PTR = F2'BYTE($NEXT'F4$) *
                            * $                            *
                            ********************************
                             .
                             .
                            *********    ***********************    ********************************
                            *IFEITH  *-+-*FIELD2 NQ LINE'PTR $ *---*LNGTH = LENGTH(TEXT'LINE) $   *
                            *********  . ***********************    *EXTRACT'TEXT = SUBSTR(TEXT'LIN*
                                       .                            *E, FIELD3, LNGTH - FIELD3 + 1)*
                                       .                            * $                            *
                                       .                            *TEXT'LINE = F2'LINE($ACCESS2(L*
********************************
*PROC GENERATE'CONTENTS'ENTRY $ *
*                               *
*                               *
********************************
 *
 *
 *
 * '' THE PROCEDURE
 *   GENERATE'CONTENTS'ENTRY
 *   OUTPUTS A A TABLE OF CONTENTS
 *   ENTRY FOR THE CURRENT MODULE.
 *   IT FINDS THE STARTING PAGE OF
 *   THE MODULE IN PAGE'REF.''
 *
 *
*******************************
*ITEM LINE INTEGER $          *
*'' LINE'NO'S DISPLACEMENT ON*
* CURRENT PAGE''              *
*ITEM LNGTH INTEGER $         *
*'' STORES STRING LENGTHS''   *
*ITEM PAGE INTEGER $          *
*'' DUMMY USED IN CALLING     *
* REMQUO''                    *
*REMQUO(LINE'NO, PAGE'LNGTH =*
* PAGE, LINE) $               *
*******************************
 *
 *
*****************     ******************************
*IF LINE EQ 0 $  *---*GENERATE'CONTENTS'HEADER $ *
*****************     ******************************
 *
 *
********************************
*TEMPC = PROC'NAME($CUR'GROUP$)*
* $                            *
*LNGTH = MIN(LENGTH(TEMPC), 30)*
* $                            *
********************************
 *
 *
****************
*IF DEBUG23 $  *    ************************************
*''7''          *---*MESSAGE = 12H(TABLE ENTRY ) $ *
*               *    *T'MESS = TEMPC $                 *
****************    *MESSAGE = CAT(MESSAGE, TEMPC) *
 *                   * $                               *
 *                   *OUT(MESSAGE, MESS'SW) $          *
 *                   ************************************
 *
********************************
*LINE = ACCESS'OUT(LINE'NO) $ *
*BYTE($MARGIN, LNGTH$)(OUT'LINE*
*($LINE$)) = BYTE($6,          *
```

```
                                                                    *INE*PTR)$) $                  *
                                       *                            *TEXT*LINE = SUBSTR(TEXT*LINE, *
                                       *                            * 1, BYTE*PTR) $               *
                                       *                            *EXTRACT*TEXT = CAT(EXTRACT*TEX*
                                       *                            *T, TEXT*LINE) $               *
                                       *                            ********************************
                                       *
                                       * ***********    ********************************
                                       +-*ORIF 1 $  *---*EXTRACT*TEXT = SUBSTR(TEXT*LIN*
                                         ***********    *E, FIELD3, BYTE*PTR - FIELD3 +*
                                                        * 1) $                         *
                                                        ********************************
```

```
* LNGTH$)(TEMPC) $               *
*DOTS(MARGIN + LNGTH + 2, 40 -   *
* LNGTH, LINE'NO) $              *
*TEMPC = IFORMAT(PAGE'REF($CUR'  *
*GFOUP$)) $                      *
*LNGTH = LENGTH(TEMPC) $         *
*TEMPC = BYTE($6,                *
* LNGTH$)(TEMPC) $               *
*BYTE($46 + MARGIN - LNGTH,      *
* LNGTH$)(OUT'LINE($LINE$)) =    *
* TEMPC $                        *
*LINE'NO = LINE'NO + 1 $         *
*RETURN $                        *
**********************************
```

DESIGN DIAGRAM OF GENERATE'CONTENTS'HEADER

```
********************************
*PROC GENERATE'CONTENTS'HEADER *
* $ PROC GENERATE'CONTENTS'HEA *
*                              *
********************************
 .
 .
 .
 .  '' THE PROCEDURE
 .   GENERATE'CONTENTS'HEADER
 .   OUTPUTS THE HEADING FOR THE
 .   TABABLE OF CONTENTS. ''
 .
 .
********************************
*ITEM CONTENTS'HEADER CHARACTER*
* $                            *
*''THE TABLE OF CONTENTS       *
* TITLE''                      *
*ITEM LNGTH INTEGER $          *
*'' STORES STRING LENGTHS''    *
********************************
 .
 .
**************
*IF DEBUG1 $   *    ********************************
*''5''         *---*MESSAGE = 15H(CONTENTS HEADER)*
*              *    * $                            *
**************      *OUT(MESSAGE, MESS'SW) $       *
 .                  ********************************
 .
*****************************
*ACVANCE'PAGE $             *
*PAGE'NO = PAGE'NO + 1 $    *
*TEMPC = IFORMAT(PAGE'NO) $ *
*LNGTH = LENGTH(TEMPC) $    *
*BYTE($PAGE'BYTE,           *
* LNGTH$)(HEADER($0$)) =    *
* BYTE($6, LNGTH$)(TEMPC) $ *
*****************************
 .
 .
**************************    ***********************    ********************************
*FOR I = 0, 1, HEAD'NO $  *---*IF I NQ NAME'INDEX $  *---*TEMPC = BYTE($6, MAXCOL -     *
**************************    ***********************    * MARGIN$)(HEADER($I$)) $      *
 .                                                       *BYTE($MARGIN, MAXCOL -        *
 .                                                       * MARGIN$)(OUT'LINE($ACCESS'OUT*
 .                                                       *(LINE'NO)$)) = TEMPC $        *
 .                                                       *LINE'NO = LINE'NO + 1 $       *
 .                                                       ********************************
 .
********************************
*LINE'NO = LINE'NO + 2 $       *
*CONTENTS'HEADER = 17H(TABLE OF*
* CONTENTS) $                  *
*TEMPC = CNVERT(CONTENTS'HEADER*
*) $                            *
*BYTE($MARGIN, 17$)(OUT*LINE($A*
*CCESS*OUT(LINE*NO)$)) =        *
* BYTE($6, 17$)(TEMPC) $        *
*LINE*NO = LINE*NO + 2 $        *
*RETURN $                       *
*********************************
```

DESIGN DIAGRAM OF GET'F1'REC

```
********************
*PROC GET'F1'REC $  *
********************
 .
 .
 .
 .  ''THE PROCEDURE GET'F1'REC
 .   TRIES TO  READ THE NEXT FILE
 .   1 RECORD. IF AN END OF FILE
 .   IS ENCOUNTERED, IT SETS
 .   EOFILE TO TRUE. ''
 .
 .
******************************
*ITEM F1'PTR INTEGER $       *
*'' THE NUMBER OF THE FILE 1 *
* RECORD ''                  *
*LAST'F1 = LAST'F1 + 1 $     *
*F1'PTR = ACCESS1(LAST'F1) $ *
******************************
 .
 .
*******************    ******************
*IF F1'PTR LS 0 $ *---*EOFILE = TRUE $ *
*******************    ******************
 .                      .
 .                      .
 .                      ***************
 .                      *IF DEBUG2 $  *    ****************************
 .                      *''5''        *---*MESSAGE = 9H(FILE1 EOF) $ *
 .                      *             *    *OUT(MESSAGE, MESS'SW) $   *
 .                      ***************    ****************************
 .                      .
 .                      .
 .                      ***********
 .                      *RETURN $ *
 .                      ***********
 .
********************************
*FILE2'INDEX = FIELD1($F1'PTR$)*
* $                            *
*STMT'TOKEN = FIELD2($F1'PTR$) *
* $                            *
*F2'RECS = FIELD3($F1'PTR$) $  *
*STMT'LNGTH = FIELD4($F1'PTR$) *
* $                            *
*RETURN $                      *
********************************
```

DESIGN DIAGRAM OF IFORMAT

```
**********************
*PROC IFORMAT(NUM) $  *
**********************
 *
 *
 *
 * ** THE FUNCTION IFORMAT
 *   ACCEPTS AN INTEGER, AND
 *   RETURNS A CHARACTER STRING
 *   (IN CONVERTED FORM)
 *   REPRESENTING THE INTEGER.**
 *
 *
********************************
*ITEM NUM INTEGER $            *
*** THE INTEGER TO BE          *
* CONVERTED**                  *
*ITEM IFORMAT CHARACTER $      *
*** THE STRING TO BE RETURNED***
*                              *
*ITEM H6 H 6 $                 *
*** A TEMPORARY FOR ENCODE**   *
*ENCODE(6H((  I6)), NUM = H6) $*
*                              *
*IFORMAT = H6 $                *
*IFORMAT = CNVERT(IFORMAT) $   *
********************************
```

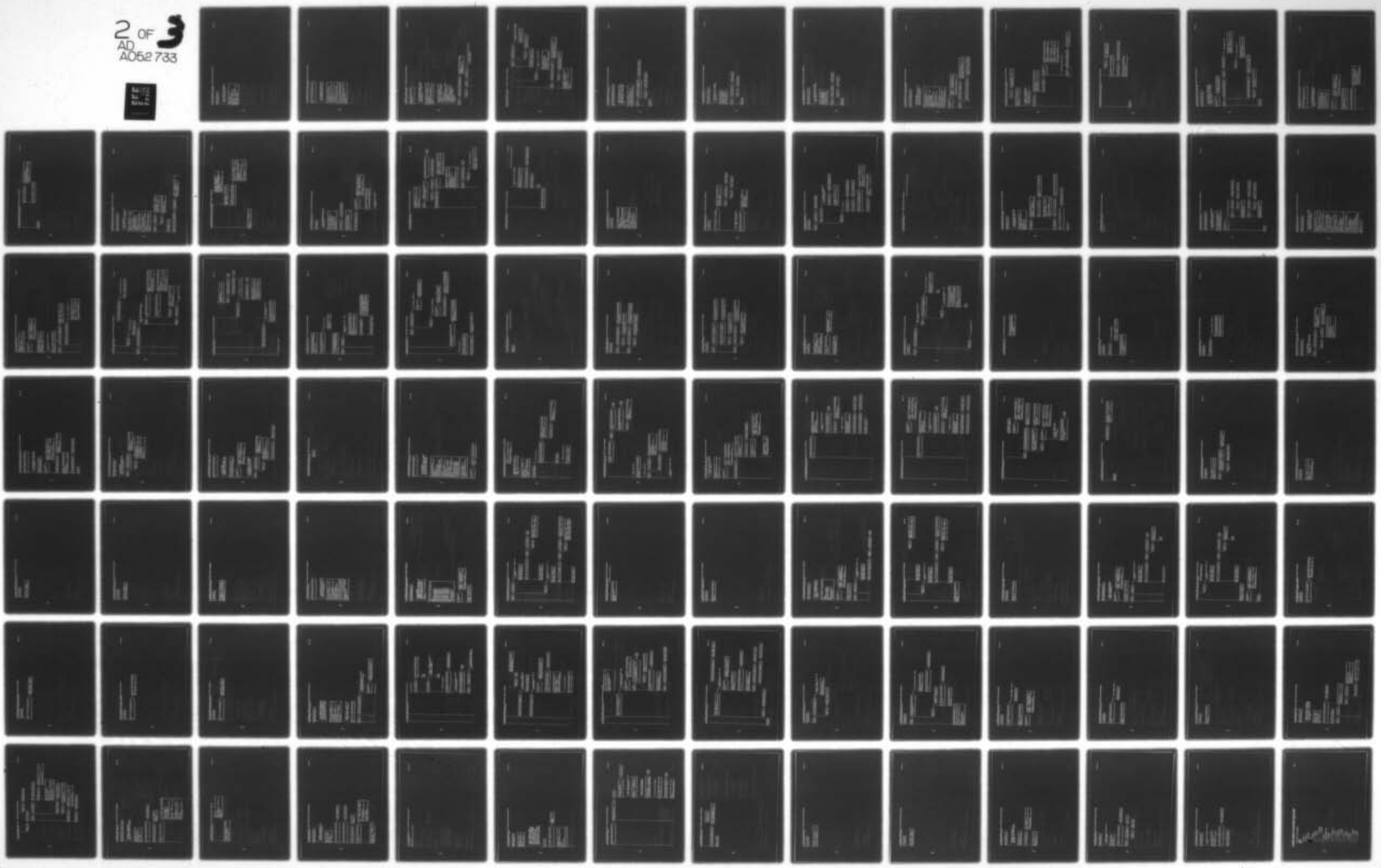
DESIGN DIAGRAM OF INIT'BLOCK'CONSTANTS

```
******************************
*PROC INIT'BLOCK'CONSTANTS $ *
******************************
 *
 *
 *
 *  ** THE PROCEDURE
 *   INIT'BLOCK'CONSTANTS IS
 *   CALLED BY PART2 TO INITIALIZE
 *   THE VARIABLES W4ICH WILL BE
 *   CONSTANT THROUGHOUT THE
 *   CREATION OF THE CODE BLOCK.
 *   **
 *
 *
********************************
*ITEM INDEX INTEGER $          *
***INDEX INTO F3'BUF OF THE    *
* CURRENT RECORD**             *
*INDEX = ACCESS3(CUR'REC) $    *
*STMT'TYPE =(/ STMT'UNIT($INDEX*
*$) /) $                       *
*TOP'LINE = START'LINE($INDEX$)*
* + DISP $                     *
*MIDPOINT = MIDPT($INDEX$) +   *
* DISP $                       *
*BCTTOM'LINE = STOP'LINE($INDEX*
*$) + CISP $                   *
*LEFT'COL = START'COL($INDEX$) *
* $                            *
*WIDTH = BLOCK'WIDTH($INDEX$) $*
*                              *
*RIGHT'COL = STOP'COL($INDEX$) *
* $                            *
*F4'START = F4'BEGIN($INDEX$) $*
*                              *
*F4'STOP = F4'END($INDEX$) $   *
*H'SON = H'PTR($INDEX$) $      *
*V'SON = V'PTR($INDEX$) $      *
*H'FATHER = BACK'H($INDEX$) $  *
*V'FATHER = BACK'V($INDEX$) $  *
*PAST'MID = FALSE $            *
********************************
 *
 *
*********    ***************************    *************************
*IFEITH  *-+-*STMT'TYPE EQ CONTROL'3 $  *---*RIGHT'POS = LEFT'COL + *
*********  * ***************************    * BOX'TAIL + WIDTH + 1 $*
 *         *                                *************************
 *         *
 *         * ***********    ********************************
 *         +-*ORIF 1 $  *---*RIGHT'POS = LEFT'COL + WIDTH +*
 *           ***********    * 1 $                          *
 *                          ********************************
 *
*****************
```

```
C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER                                                PAGE 91
DESIGN DIAGRAM OF INIT'BLOCK'CONSTANTS

*SKIP = FALSE $ *
*****************
 *
 *
******************
*IF V'SON GR 0 $  *----+
******************     * ** GET V'SON'S TOP LINE IN
 *                     *   CASE OF BUFFER RUNOVER **
 *                     *
 *                     *
 *                    ********************************
 *                    *SON'TOP = START'LINE($ACCESS3(*
 *                    *V'SON)$) + DISP $             *
 *                    ********************************
 *
***********
*RETURN $ *
***********
```

AD-A052 733   CHARLES STARK DRAPER LAB INC CAMBRIDGE MA   F/G 9/2
JOVIAL STRUCTURED DESIGN DIAGRAMMER (JSDD). VOLUME III. PROGRAM--ETC(U)
FEB 78 G GODDARD, M WHITWORTH, E STROVINK   F30602-76-C-0408
UNCLASSIFIED   R-1120-VOL-3-PT-3   RADC-TR-78-9-VOL-3-PT-3   NL

2 OF 3
AD
A052733



```
DESIGN DIAGRAM OF INITIALIZE

********************
*PROC INITIALIZE $  *
********************
 *
 *
 *
 * '' THE PROCEDURE INITIALIZE
 *   SETS UP THE INITIAL VALUES
 *   FOR DOG EXECUTION. ''
 *
 *
**********************************
*SINGLE'SPACE = NOT              *
* DOUBLE'SPACE $                 *
*LAST'LINE = HEADROOM $          *
*LOW'LIM = ST'MAX / 2 $          *
*MAX'WIDTH = ST'MAX + 10 $       *
*PAGE'WIDTH = PAGE'WIDTH - 1 $   *
*OPEN INPUT FILE1 $              *
*OPEN INPUT FILE2 $              *
**********************************
```

```
DESIGN DIAGRAM OF INITIATE*RECORD

*********************************
*PROC INITIATE*RECORD(INIT*REC) *
* $                               *
*                               *
*********************************
 *
 *
 *
 * ** THE PROCEDURE INITIATE*REC
 *   SETS STARTING VALUES IN A
 *   NEWLY CREATED FILE 3 RECORD.
 *   **
 *
 *
*********************************
*ITEM INIT*REC INTEGER $        *
*** REC NO. OF BLOCK BEING      *
* INITIATED. **                 *
*F4*BEGIN($ACCESS3(INIT*REC)$) *
* = FIRST*4 $                   *
*F4*END($ACCESS3(INIT*REC)$) = *
* LAST*4 $                      *
*LINES($ACCESS3(INIT*REC)$) =  *
* LINES*OUT($ACCESS4(FIRST*4)$)*
* $                             *
*BLOCK*WIDTH($ACCESS3(INIT*REC)*
*$) = MAX*LINE*LNGTH($ACCESS4(F*
*IRST*4)$) $                    *
*STMT*UNIT($ACCESS3(INIT*REC)$)*
* = BOX*MAP(STMT*TOKEN) $       *
*WRITE3 = TRUE $                *
*RETURN $                       *
*********************************
```

```
DESIGN DIAGRAM OF INSERT

*************************
*PROC INSERT(GROUP'TYPE, *
* F3'INDEX) $           *
*                       *
*************************
 .
 .
 .
 . '' THE PROCEDURE INSERT  LINKS
 .   AN ENTRY INTO THE GROUP TABLE
 .   FOR A TREE WHICH IS BEING
 .   INITITATED.''
 .
 .
******************************
*ITEM F3'INDEX INTEGER $     *
*''INDEX OF FILE 3 RECORD BEING*
* INSERTED''                 *
*ITEM GROUP'TYPE INTEGER $   *
*''TYPE IS MODULE HEAD OR STUMP*
* HEAD''                     *
******************************
 .
 .
 .
 . '' THIS PROC INSERTS THE NEW
 .   GROUP INTO THE GROUP ''
 . '' STORE. A GROUP IS EITHER A
 .   STUMP OR A PROC (OR CLOSE,
 .   ETC) ''
 . '' DECLARATION. ''
 . '' THE GROUP STORE IS A LINKED
 .   LIST OF GROUPS ORDERED ''
 . '' APPROPRIATELY FOR OUTPUT
 .   PROCESSING.''
 . '' ALTHOUGH UNECESSARY FOR
 .   JOVIAL, NESTED PROCS ''
 . ''COULD BE ACCOMODATED BY
 .   INTRODUCING A GROUP
 .   ENVIRONMENT ''
 . '' STACK. ''
 .
 .
*********     ***************************     ***************************
*IFEITH  *-+-*GROUP'AVAIL GR GROUP'MAX $  *---*MESSAGE = 20H(GROUP STORE*
*********  . ***************************     * OVERFLOW) $              *
           .                                 *PH2ERR(MESSAGE) $         *
           .                                 ***************************
           .
           . ***********     *********     **************
           +-*ORIF 1 $  *---*IFEITH  *-+-*PROC'FLAG $  *----+
             ***********     *********  . **************     . ''WE HAVE EITHER A NEW PROC OR
                                 .      .                    .  A STUMP HANGING ''
                                 .      .                    . '' FROM A PROC .''
                                 .      .                    .
                                 .      .                    .
 .         .                   **********   ********************
 .         .                   *IFEITH  *-+-*LAST*PROC LS 0 $  *----+
 .         .                   ********** . ********************    .  ** THIS IS THE FIRST PROC
 .         .                    .         .                         .   ENCOUNTERED** ** THIS IS
 .         .                    .         .                         .
 .         .                    .         .                         .
 .         .                    .         .                        **********************
 .         .                    .         .                        *NEXT($_AST*STUMP$) =*
 .         .                    .         .                        * GROUP*AVAIL $      *
 .         .                    .         .                        **********************
 .         .                    .         .
 .         .                    .         . ************   *********************
 .         .                    .         +-*ORIF 1 $  *---*NEXT($LAST*PRO:$) =*
 .         .                    .           ************   * GROUP*AVAIL $     *
 .         .                    .                          *********************
 .         .                    .
 .         .                    .
 .         .                   ****************************
 .         .                   *NEXT($GROUP*AVAIL$) = 0 $ *
 .         .                   *LAST*PROC = GROUP*AVAIL $ *
 .         .                   ****************************
 .         .
 .         . ************
 .         +-*ORIF 1 $  *----+
 .           ************    . ** WE HAVE A MAIN PROGRAM
 .                           .  STUMP **
 .                           .
 .                          *****************************
 .                          *NEXT($GROUP*AVAIL$) =      *
 .                          * NEXT($LAST*STUMP$) $      *
 .                          *NEXT($LAST*STUMP$) =       *
 .                          * GROUP*AVAIL $             *
 .                          *LAST*STUMP = GROUP*AVAIL $ *
 .                          *****************************
**********   *****************************   ********************************
*IFEITH  *-+-*GROUP*TYPE EQ PROC*ROOT $  *---*PROC*NAME($GROUP*AVAIL$) =    *
********** . *****************************   * EXTRACT*NAME() $             *
 .         .                                 *PROC*COUNT = PROC*COUNT + 1 $ *
 .         .                                 ********************************
 .         .
 .         . ************   ****************************
 .         +-*ORIF 1 $  *---*PROC*NAME($GROUP*AVAIL$) =*
 .           ************   * SPACES(MAXCOL) $         *
 .                          ****************************
 .
*******************************
*CUR*GROUP = GROUP*AVAIL $    *
*F3*REC($GROUP*AVAIL$) =      *
* F3*INDEX $                  *
*GROUP*AVAIL = GROUP*AVAIL + 1*
* $                           *
*RETURN $                     *
*******************************
```

DESIGN DIAGRAM OF LEGAL'STUMP

```
********************************
*PROC LEGAL'STUMP(STMT'TYPE) $  *
********************************
 *
 *
 *
 * '' THE FUNCTION LEGAL STUMP
 *   ACCEPTS A CODE BLOCK TYPE AS
 *   INPUT AND OUTPUTS A 1 IF THE
 *   CODE BLOCK IS A  LEGAL STUMP
 *   ROOT. OTHERWISE. IT RETURNS A
 *   0. ''
 *
 *
********************************
*ITEM LEGAL'STUMP B $          *
*'' HOLDS THE RETURN VALUE''   *
*ITEM STMT'TYPE INTEGER $      *
*''THE INPUT CODE BLOCK TYPE'' *
********************************
 *
 *
*********    **************************
*IFEITH  *-+-*STMT'TYPE EQ CONTROL'3 OR *   ************************
*********  * * STMT'TYPE EQ COMMENT'2 $  *---*LEGAL'STJMP = FALSE $ *
 *         * *                          *   ************************
 *         * **************************
 *         *
 *         *
 *         * ***********   ************************
 *         +-*ORIF 1 $  *---*LEGAL'STUMP = TRUE $ *
 *           ***********   ************************
 *
***********
*RETURN $ *
***********
```

```
C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER                                   PAGE 97
DESIGN DIAGRAM OF MAX

**************************
*PROC MAX(INT1, INT2) $  *
**************************
 *
 *
 *
 * ** THE FUNCTION MAX RETURNS
 *   THE LARGER OF THE TWO VALUES
 *   PASSED TO IT. **
 *
 *
**********************
*ITEM INT1 INTEGER $ *
***INPUT PARAMETER** *
*ITEM INT2 INTEGER $ *
***INPUT PARAMETER** *
*ITEM MAX INTEGER $  *
*** RETURN VALUE**   *
**********************
 *
 *
**********   ******************   ***************
*IFEITH  *-+-*INT2 GR INT1 $  *---*MAX = INT2 $ *
********** * ******************   ***************
 *         *
 *         *
 *         * ************   ***************
 *         +-*ORIF 1 $  *---*MAX = INT1 $ *
 *           ************   ***************
 *
************
*RETURN $ *
************
```

```
C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER                                   PAGE 98
DESIGN DIAGRAM OF MIN

*************************
*PROC MIN(INT1, INT2) $  *
*************************
 *
 *
 *
 * "" THE FUNCTION MIN RETURNS
 *   THE LESSER OF THE TWO VALUES
 *   PASSED TO IT. ""
 *
 *
***********************
*ITEM INT1 INTEGER $  *
*""INPUT PARAMETER""  *
*ITEM INT2 INTEGER $  *
*""INPUT PARAMETER""  *
*ITEM MIN INTEGER $   *
*""THE RETURN VALUE"" *
***********************
 *
 *
*********  ******************    ***************
*IFEITH  *-+-*INT2 LS INT1 $  *---*MIN = INT2 $ *
********* * ******************    ***************
 *        *
 *        *
 *        * ***********     ***************
 *        +-*ORIF 1 $  *---*MIN = INT1 $ *
 *          ***********     ***************
 *
***********
*RETURN $ *
***********
```

```
***************************
*PROC OUTPUT'BOX'BOTTOM $  *
***************************
 *
 *
 *
 * ''THE PROCEDURE
 *  OUTPUT'BOX'BOTTOM OUTPUTS THE
 *  BOTTOM OF THE CODE BLOCK. IT
 *  ALSO PERFORMS SOME DOUBLE
 *  BUFFERING OPTIMIZATION. ''
 *
 *
********************************
*ITEM QUO1 INTEGER $           *
*''THE PAGE ON WHICH THE BOX   *
* BOTTOM APPEARS''             *
*ITEM QUO2 INTEGER $           *
*''THE PAGE ON WHICH V'SON'S   *
* BOX TOP APPEARS''            *
*ITEM REM1 INTEGER $           *
*''DUMMY USED IN CALLING       *
* REMQUO''                     *
*ITEM REM2 INTEGER $           *
*'' DUMMY USED IN CALLING      *
* REMQUO''                     *
*ITEM X1 INTEGER $             *
*''COLUMN OF CONNECTING LINE'' *
*ITEM Y1 INTEGER $             *
*''START LINE OF CONNECTOR''   *
*ITEM Y2 INTEGER $             *
*'' END LINE OF CONNECTOR''    *
********************************
 *
 *
**************
*IF DEBUG9 $ *    ********************************
*''7''       *---*MESSAGE = 11H(BOTTOM OUT ) $  *
*            *   *T'MESS = IFORMAT(CUR'REC) $   *
**************   *MESSAGE = CAT(MESSAGE, T'MESS)*
 *               * $                            *
 *               *OUT(MESSAGE, MESS'SW) $       *
 *               ********************************
 *
*********  **************************      ********************************
*IFEITH  *-+-*STMT'TYPE EQ COMMENT'2 $  *---*BYTE($LEFT'COL + 1,         *
*********  * **************************     * 1$)(OUT'LINE($ACCESS'OUT(BOTT*
 *         *                                *OM'LINE)$)) = 1H(') $        *
 *         *                                ********************************
 *         *
 *         * ********************************    ****************************
 *         +-*ORIF STMT'TYPE EQ CONTROL'3 $  *---*STARS(BOX'TAIL + LEFT'COL,*
 *         * ********************************    * WIDTH + 2, BOTTOM'LINE) $*
 *         *                                     ****************************
 *         *
 *         *
```

```
C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER                                                              PAGE 100
DESIGN DIAGRAM OF OUTPUT'BOX'BOTTOM

 *         * ************   *****************************
 *         +-*ORIF 1 $  *---*STARS(LEFT'COL, WIDTH + 2,*
 *           ************   * BOTTCM'LINE) $            *
 *                          *****************************
 *
*****************************
*LINE'NO = BOTTOM'LINE $    *
*LAST'LINE = MAX(LAST'LINE,*
* LINE'NO) $                *
*****************************
 *
 *
 *
 * '' ATTEMPT TO REDUCE THE
 *  NUMBER OF INVOCATIONS OF   ''
 * ''TRANSFER'WRITE'OUT BY
 *  DRAWING PART OF ''
 * '' CONNECTING LINE NOW IF A
 *  BUFFER SPAN OCCURS. ''
 *
 *
*******************    *********************************
*IF V'SON GR 0 $ ' *---*REMQUO(LINE'NO, OUT'BUF'SIZE =*
*******************    * QUO1, REM1) $                 *
 *                     *REMQUO(SON'TOP, OUT'BUF'SIZE =*
 *                     * QUO2, REM2) $                 *
 *                     *********************************
 *                      *
 *                      *
 *                     *********************    ***************
 *                     *IF QUO1 NQ QUO2 $  *---*IF DEBUG24 $  *    **********************************
 *                     *********************    *''11''       *---*MESSAGE = 15H(AVOID BUF SPAN )*
 *                                              *             *    * $                              *
 *                                              ***************    *T'MESS = IFORMAT(CUR'REC) $     *
 *                                               *                 *MESSAGE = CAT(MESSAGE, T'MESS)*
 *                                               *                 * $                              *
 *                                               *                 *T'MESS = 5H( AND ) $            *
 *                                               *                 *MESSAGE = CAT(MESSAGE, T'MESS)*
 *                                               *                 * $                              *
 *                                               *                 *T'MESS = IFORMAT(V'SON) $       *
 *                                               *                 *MESSAGE = CAT(MESSAGE, T'MESS)*
 *                                               *                 * $                              *
 *                                               *                 *OUT(MESSAGE, MESS'SW) $         *
 *                                               *                 **********************************
 *                                               *
 *                                               *
 *                                               * '' A BUFFER SPAN DOES OCCUR ''
 *                                               *
 *                                               *
 *                                               *
 *                                              **********    ***************************     ***********************
 *                                              *IFEITH  *-+-*STMT'TYPE EQ CONTROL'3 $ *---*X1 = LEFT'COL $      *
 *                                              **********  * ***************************     *Y1 = TOP'LINE + 1 $ *
 *                                               *          *                                 ***********************
 *                                               *          *
 *                                               *          *
```

DESIGN DIAGRAM OF OUTPUT'BOX'BOTTOM

```
 *                                            *         * ***********   **********************
 *                                            *         +-*ORIF 1 $  *---*X1 = LEFT'COL + 1 $ *
 *                                            *           ***********   *Y1 = BOTTOM'LINE $  *
 *                                            *                         **********************
 *                                            *
 *                                           ********************************
 *                                           *Y2 = (QUO1 + 1) * OUT'BUF'SIZE*
 *                                           * $                            *
 *                                           ********************************
 *                                            *
 *                                            *
 *                                           *********************   **********************************
 *                                           *IF Y2 - Y1 GR 1 $  *---*DRAW'LINE(X1, Y1, X1, Y2) $ *
 *                                           *********************   **********************************
 *                                            *
 *                                            *
 *                                           *******************************
 *                                           *STOP'LINE($ACCESS3(CUR'REC)$)*
 *                                           * = Y2 - DISP - 1 $           *
 *                                           *WRITE3 = TRUE $              *
 *                                           *******************************
 *
***********
*RETURN $ *
***********
```

```
C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER
DESIGN DIAGRAM OF OUTPUT'BOX'TOP


*************************
*PROC OUTPUT'BOX'TOP $  *
*************************
 .
 .
 .
 .  ''THE PROCEDURE OUTPUT'BOX'TOP
 .   OUTPUTS THE TOP LINE OF THE
 .   CODE BLOCK. DRAW THE TOP LINE
 .   OF BOX -- NO NEED TO  HANDLE
 .   HEADINGS HERE ''
 .
 .
****************
*IF DEBUG17 $  *    **************************
*''5''          *---*MESSAGE = 7H(TOP OUT) $ *
*              *    *OUT(MESSAGE, MESS'SW) $ *
****************    **************************
 .
 .
**********   ****************************   **********   *******************
*IFEITH  *-+-*STMT'TYPE EQ COMMENT'2 $  *---*IFEITH  *-+-*H'FATHER NQ 0 $  *-----+
********** . ****************************   ********** . *******************     . '' HORIZONTAL ENTRY ''
 .         .                                           .                         .
 .         .                                           .                         .
 .         .                                           .                        ********************************
 .         .                                           .                        *BYTE($LEFT'COL,               *
 .         .                                           .                        * 2$)(OUT'LINE($ACCESS'OUT(TOP'*
 .         .                                           .                        *LINE)$)) = 2H(-+) $           *
 .         .                                           .                        ********************************
 .         .                                           .
 .         .                                           . ************
 .         .                                           +-*ORIF 1 $  *----+
 .         .                                             ************    . '' VERTICAL ENTRY ''
 .         .                                                             .
 .         .                                                             .
 .         .                                                            ********************************
 .         .                                                            *BYTE($LEFT'COL + 1,           *
 .         .                                                            * 1$)(OUT'LINE($ACCESS'OUT(TOP'*
 .         .                                                            *LINE)$)) = 1H(') $            *
 .         .                                                            ********************************
 .         .
 .         .
 .         . *********************************   ****************************
 .         +-*ORIF STMT'TYPE EQ CONTROL'3 $  *---*STARS(BOX'TAIL + LEFT'COL,*
 .         . *********************************   * WIDTH + 2, TOP'LINE) $   *
 .         .                                     ****************************
 .         .
 .         . ************   ****************************
 .         +-*ORIF 1 $  *---*STARS(LEFT'COL, WIDTH + 2,*
 .           ************   * TOP'LINE) $              *
 .                          ****************************
 .
***********
*RETURN $ *
***********
```

```
C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER                                                     PAGE 103
DESIGN DIAGRAM OF OUTPUT'HEADER

********************************
*PROC OUTPUT'HEADER(PAGE'TOP) $ *
*                              *
*                              *
********************************
 .
 .
 .
 .
 .  ** THE PROCEDURE OUTPUT'HEADER
 .   OUTPUTS THE PAGE HEADING
 .   STARTING ON THE LINE PASSED
 .   TO IT. **
 .
 .
*****************************
*ITEM LNGTH INTEGER $       *
*** STORES STRING LENGTHS** *
*ITEM PAGE'TOP INTEGER $    *
*** THE LINE ON WHICH THE   *
* HEADING STARTS**          *
*ITEM TEMP'LINE CHARACTER $ *
*** A TMP FOR CHARACTER     *
* STRINGS**                 *
*****************************
 .
 .
***************
*IF DEBUG17 $  *   *******************************
***7**         *---*T'MESS = 9H(PAGE'TOP ) $     *
*              *   *MESSAGE = IFORMAT(PAGE'TOP) $ *
***************    *MESSAGE = CAT(T'MESS, MESSAGE)*
 .                 * $                            *
 .                 *OUT(MESSAGE, MESS'SW) $       *
 .                 *******************************
 .
******************************
*TEMP'LINE = OUT'LINE($ACCESS'O*
*UT(PAGE'TOP)$) $             *
*TEMP'LINE = CNVERT(TEMP'LINE) *
* $                           *
******************************
 .
 .
******************************
*IF TEMP'LINE EQ SPACES(MAXCOL) *    ****************************
* $                             *---*PAGE'NO = PAGE'NO + 1 $     *
*                               *   *TEMPC = IFORMAT(PAGE'NO) $ *
******************************      *LNGTH = LENGTH(TEMPC) $    *
 .                                  *BYTE($PAGE'BYTE,           *
 .                                  * LNGTH$)(HEADER($0$)) =    *
 .                                  * BYTE($6, LNGTH$)(TEMPC) $ *
 .                                  ****************************
 .                                   .
 .                                   .
 .                                   .
 .                                   .
```

```
C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER
DESIGN DIAGRAM OF OUTPUT'HEADER

 .                                   **************************    ********************************
 .                                   *FOR I = 0, 1, HEAD'NO $  *---*TEMPC = BYTE($6,              *
 .                                   **************************    * MAXCOL$)(HEADER($I$)) $       *
 .                                    .                            *BYTE($MARGIN, MAXCOL$)(OUT'LIN*
 .                                    .                            *E($ACCESS'OUT(PAGE'TOP + I)$))*
 .                                    .                            * = TEMPC $                    *
 .                                    .                            ********************************
 .                                    .
 .                                   ********************************
 .                                   *LINE'NO = LINE'NJ + HEADROOM $*
 .                                   *                              *
 .                                   ********************************
 .
***********
*RETURN $ *
***********
```

```
DESIGN DIAGRAM OF OUTPUT'TITLE'PAGE

***************************
*PROC OUTPUT'TITLE'PAGE $  *
***************************
 *
 *
 *
 * '' THE PROCEDURE
 *  OUTPUT'TITLE'PAGE OUTPUTS THE
 *  TITLE PAGE OF THE DIAGRAM. ''
 *
 *
********************************
*ITEM LINE INTEGER $           *
*''DISPLACEMENT ON THE CURRENT*
* PAGE''                       *
*ITEM LNGTH INTEGER $          *
*'' STORES STRING LENGTHS ''   *
*ITEM PAGE INTEGER $           *
*''DUMMY USED IN CALLING       *
* REMQUO''                     *
*ITEM START'TITLE INTEGER $    *
*''PAGE LINE ON WHICH TITLE    *
* STARTS''                     *
*ITEM TEMP'LINE CHARACTER $    *
*'' A CHARACTER TEMP''         *
*ITEM TITLE'INDEX INTEGER $    *
*'' THE INDEX INTO THE TITLE   *
* ARRAY''                      *
*ITEM TITLE'PAGES INTEGER $    *
*''NUMBER OF PAGES SPANNED BY  *
* TITLES''                     *
********************************
 *
 *
*********    *************    ***************************
*IFEITH  *-+-*HEADING $  *---*TITLE'PAGES = TITLE'NO /*
*********  * *************    * (PAGE'LNGTH - 2) $     *
 *         *                  *START'TITLE = 2 $       *
 *         *                  **************************
 *         *
 *         * ***********    **************************
 *         +-*ORIF 1 $  *---*TITLE'PAGES = TITLE'NO /*
 *           ***********    * (PAGE'LNGTH) $         *
 *                          *START'TITLE = 0 $       *
 *                          **************************
 *
*********************
*TITLE'INDEX = 0 $ *
*********************
 *
 *
******************************    ****************    ********************************
*FOR I = 0, 1, TITLE'PAGES $  *---*IF HEADING $  *---*TEMPC = IFORMAT(I + 1) $       *
******************************    ****************    *LNGTH = LENGTH(TEMPC) $        *
 *                                 *                  *BYTE($PAGE'BYTE,               *
 *                                 *                  * LNGTH$)(HEADER($0$)) =        *
```

```
DESIGN DIAGRAM OF OUTPUT'TITLE'PAGE
                                                      * BYTE($6, LNGTHS)(TEMPC) $    *
 *                                  *                 *TEMP'LINE = BYTE($6,          *
 *                                  *                 * MAXCOLS)(HEADER($0$)) $      *
 *                                  *                 *BYTE($MARGIN, MAXCOLS)(OUT'LIN*
 *                                  *                 *E($ACCESS'OUT(LINE'NO)$)) =   *
 *                                  *                 * TEMP'LINE $                  *
 *                                  *                 *LINE'NO = LINE'NO + 2 $       *
 *                                  *                 ********************************
 *                                  *
 *                                 ******************************
 *                                 *REMQUO(LINE'NO, PAGE'LNGTH =*
 *                                 * PAGE, LINE) $             *
 *                                 ******************************
 *                                  *
 *                                  *
 *                                 ********************************
 *                                 *DO WHILE [LINE GQ START'TITLE *   ********************************
 *                                 * AND TITLE'INDEX LQ TITLE'NO) *---*TEMP'LINE = BYTE($6,          *
 *                                 *                              *   * MAXCOLS)(TITLE($TITLE'INDEX$)*
 *                                 **p*****************************   *) $                           *
 *                                                                    *BYTE($MARGIN, MAXCOLS)(OUT'LIN*
 *                                                                    *E($ACCESS'OUT(LINE'NO)$)) =   *
 *                                                                    * TEMP'LINE $                  *
 *                                                                    *LINE'NO = LINE'NO + 1 $       *
 *                                                                    *TITLE'INDEX = TITLE'INDEX + 1 *
 *                                                                    * $                            *
 *                                                                    *REMQUO(LINE'NO, PAGE'LNGTH =  *
 *                                                                    * PAGE, LINE) $                *
 *                                                                    ********************************
 *
******************************
*PAGE'NO = TITLE'PAGES + 1 $ *
*RETURN $                    *
******************************
```

```
DESIGN DIAGRAM OF PART2

***************
*PROC PART2 $  *
***************
 *
 *
 *
 * ** THE PROCEDURE PART2 IS THE
 *   DRIVER ROUTINE FOR PART2 OF
 *   THE DDG. **
 *
 *
*****************************
*ITEM LNGTH INTEGER $        *
*** STORES STRING LENGTHS** *
*ITEM TMPI1 INTEGER $        *
*** AN INTEGER TEMP**        *
*ITEM TMPI2 INTEGER $        *
*** AN INTEGER TEMP**        *
*PART2'INIT $                *
*****************************
 *
 *
****************    **********************
*IF TITLE'SW $  *---*OUTPUT'TITLE'PAGE $ *
****************    **********************
 *
 *
*******************************
*COMPUTE'PAGE'NUMBERS $       *
*CUR'GROUP = NEXT($0$) $      *
*** 1ST GROUP IN STORE -- MAIN*
* PGM **                      *
*******************************
 *
 *
****************************    ***************
*DO WHILE [CUR'GROUP GR 0]  *---*IF DEBUG18 $  *     ********************************
****************************    ***7**         *---*MESSAGE = 12H(START GROUP ) $ *
                                *              *    *T'MESS = IFORMAT(CUR'GROUP) $ *
                                ***************     *MESSAGE = CAT(MESSAGE, T'MESS)*
                                 *                  * $                            *
                                 *                  *OUT(MESSAGE, MESS'SW) $       *
                                 *                  ********************************
                                 *
                                ******************************
                                *CUR'REC = F3'REC($CUR'GROUP$)*
                                * $                           *
                                *** 1ST BLOCK IN CUR'GROUP ** *
                                ******************************
                                 *
                                 *
                                 *
                                 *  ** INITIALIZE TRAVERSAL
                                 *  STRUCTURE **
                                 *
                                 *
```

```
*********************************
*TRAVERSE'TOP = 0 $             *
*TRAVERSE'STACK($TRAVERSE'TOP$)*
* = 0 $                         *
*ADVANCE'PAGE $                 *
*********************************
 *
 *
*********    ****************************
*IFEITH  *-+-*PROC'NAME($CUR'GROUP$) NQ *     **********************************
*********  * * SPACES(MAXCOL) $          *---*IF HEADING AND NAME'INDEX GQ 0 *   ******
 *         * *                          *    * $                              *--* 110*
 *         * ****************************    *                              *    ******
 *         *                                 **********************************
 *         *
 *         * ***********   ***************   *****************************
 *         +-*ORIF 1 $ *---*IF HEADING $ *---*OUTPUT'HEADER(LINE'NO) $ *
 *           ***********   ***************   *****************************
 *
****************************   ****************
*DO WHILE [CUR'REC GR 0]  *---*IF DEBUG19 $  *     *********************************
****************************   *''7''          *---*MESSAGE = IFORMAT(CUR'REC) $  *
 *                              *               *    *T'MESS = 4H(REC ) $          *
 *                              ****************     *MESSAGE = CAT(T'MESS, MESSAGE)*
 *                               *                   * $                            *
 *                               *                   *OUT(MESSAGE, MESS'SW) $       *
 *                               *                   *********************************
 *                               *
 *                              ***********************************
 *                              *TRAVERSE'TOP = TRAVERSE'TOP +  *
 *                              * 1 $                            *
 *                              *TRAVERSE'STACK($TRAVERSE'TOP$)*
 *                              * = CUR'REC $                    *
 *                              *INIT'BLOCK'CONSTANTS $          *
 *                              *CONNECT'BOXES $                 *
 *                              *OUTPUT'BOX'TOP $                *
 *                              *LINE'NO = TOP'LINE + 1 $        *
 *                              *'' POSITION FOR 1ST TEXT LINE  *
 *                              * ''                             *
 *                              ***********************************
 *                               *
 *                               *
 *                              *********   ********************    ******
 *                              *IFEITH  *-+-*STMT'TYPE NQ 0 $  *--* 111*
 *                              *********  * ********************    ******
 *                               *         *
 *                               *         *
 *                               *         * ***********
 *                               *         +-*ORIF 1 $  *----+
 *                               *           ***********     * '' WE HAVE A NULL BOX ''
 *                               *                           *
 *                               *                           *
 *                               *                          ***********************************
 *                               *                          *NEW'TEXT = 1H(-) $               *
 *                               *                          *NEW'TEXT = INVERT(NEW'TEXT) $   *
 *                               *                          *CONSTRUCT'LINE $                 *
```

```
                                                                                          *******************************
                                 .                            .
                                 .                            .
                                 .                            .  ** SEE IF WE NEED A HEADING
                                 .                            .    HERE **
                                 .                            .
                                 .                            .
                                 .                           ******************************
                                 .                           *REMQUO(LINE'NO, PAGE'LNGTH =*
                                 .                           * TMPI1, TMPI2) $            *
                                 .                           ******************************
                                 .                            .
                                 .                            .
                                 .                           ******************************    **************************
                                 .                           *IF TMPI2 EQ 0 AND HEADING $ *----*OUTPUT'HEADER(LINE'NO) $ *
                                 .                           ******************************    **************************
                                 .                            .
                                 .                           *********************
                                 .                           *OUTPUT'BOX'BOTTOM $ *
                                 .                           *********************
                                 .                            .
                                 .                            .
                                 .                           ******
                                 .                           * 112*
                                 .                           ******
                                *******************************
                                *CUR'GROUP = NEXT($CUR'GROUPS)*
                                * $                           *
                                *******************************
```

DESIGN DIAGRAM OF PART2

```
****************
* 110 FROM 108 *
****************
 *
 *
 *
**********************************
*LNGTH = LENGTH(PROC'NAME($CUR'*
*GROUP$)) $                     *
*BYTE($NAME'BYTE,               *
* LNGTH$)(HEADER($NAME'INDEX$))*
* = BYTE($6, LNGTH$)(PROC'NAME(*
*$CUR'GROUP$)) $                *
*TMPI1 = MAXCOL - (NAME'BYTE +  *
* LNGTH - 1) $                  *
*TMPI2 = NAME'BYTE + LNGTH $    *
*TEMPC = SPACES(TMPI1) $        *
*BYTE($TMPI2, TMPI1$)(HEADER($N*
*AME'INDEX$)) = BYTE($6,        *
* TMPI1$)(TEMPC) $              *
*OUTPUT'HEADER(LINE'NO) $       *
*PAGE'REF($CUR'GROUP$) =        *
* PAGE'NO $                     *
**********************************
```

```
****************
* 111 FROM 108 *
****************
 *
 *
 *
*********************************
*FOR I = F4'START, 1, F4'STOP $ *      *******************************
*                               *---*NEW'TEXT = EXTRACT'TEXT(I) $ *
*********************************      *CONSTRUCT'LINE $              *
 *                                     *LINE'NO = LINE'N) + 1 $      *
 *                                     *******************************
 *                                      *
 *                                      *
 *                                     *********     ****************     ***************
 *                                     *IFEITH  *-+-*DOUBLE'SPACE $  *---*SKIP = TRUE $ *
 *                                     *********  *  ****************     ***************
 *                                                *
 *                                                *  **********     ***************
 *                                                +-*ORIF 1 $  *---*SKIP = FALSE $ *
 *                                                   **********     ***************
 *
 * ** IF A BLANK LINE HAS BEEN
 *   LEFT, INSERT BOX SIDES **
 *
********************************
*IF LINE'NO EQ BOTTOM'LINE - 1 *    ******************************
*  $                           *---*NEW'TEXT = SPACES(1) $    *
*                              *    *SKIP = FALSE $               *
********************************    *CONSTRUCT'LINE $             *
                                    *LINE'NO = LINE'NO + 1 $     *
                                    ******************************
```

```
****************
* 112 FROM 109 *
****************
 .
 .
 .
**********   ***************   ********************
*IFEITH  *-+-*H'SON GR 0 $ *---*CUR'REC = H'SON $ *
********** . ***************   ********************
           .
           . ********************
           +-*ORIF V'SON GR 0 $ *-----+
           . ********************     . ** IF THERE'S A HORIZ STUMP
           .                          .   AND A NONSTUMP VSON **
           .                          . ** CATCH IT HERE . **
           .                          .
           .                         ******************   *************************
           .                         *IF H'SON LS 0 $ *---*DISPLAY'STUMP'REF(1) $ *
           .                         ******************   *************************
           .                          .
           .                          .
           .                         *******************************
           .                         *CUR'REC = V'SON $            *
           .                         *TRAVERSE'TOP = TRAVERSE'TOP -*
           .                         * 1 $                         *
           .                         *******************************
           .
           . ***********
           +-*ORIF 1 $ *-----+
             ***********     . ** A STUMP **
                             .
                             .
                            ******************    *************************
                            *IF H'SON LS 0 $ *----*DISPLAY'STUMP'REF(1) $ *
                            ******************    *************************
                             .
                             .
                            ******************    *************************
                            *IF V'SON LS 0 $ *----*DISPLAY'STUMP'REF(0) $ *
                            ******************    *************************
                             .
                             .
                            *********************************
                            *DO WHILE [V'PTR($ACCESS3(CUR'R *    ********************************
                            *EC)$) LQ 0 AND CUR'REC GR 0]   *----*TRAVERSE'TOP = TRAVERSE'TOP - *
                            *                               *    * 1 $                          *
                            *********************************    *CUR'REC = TRAVERSE'STACK($TRAV*
                             .                                   *ERSE'TOP$) $                  *
                             .                                   ********************************
                             .
                            ********************    ********************************
                            *IF CUR'REC GR 0 $ *----*CUR'REC = V'PTR($ACCESS3(CUR'R*
                            ********************    *EC)$) $                       *
                                                    *TRAVERSE'TOP = TRAVERSE'TOP - *
                                                    * 1 $                          *
```

```
********************************
```

DESIGN DIAGRAM OF PART2'INIT

```
********************
*PROC PART2'INIT $ *
********************
 *
 *
 *
 * ** THE PROCEDURE PART2'INIT
 *  PERFORMS SET-UP TASKS FOR
 *  PART2 EXECUTION. **
 *
*****************************
*ITEM LNGTH INTEGER $        *
*** STORES STRING LENGTHS** *
*LINE'NO = 0 $              *
*LAST'LINE = 0 $            *
*READSW = TRUE $            *
*PAGE'NO = 0 $              *
*****************************
 *
 *
***************   *******************************
*IF HEADING $ *---*DISPLAY'WIDTH = 16 $         *
***************   *HEADROOM = HEAD'NO + 2 $     *
 *                *PAGE'BYTE = PAGE'WIDTH - 5 $ *
 *                *******************************
 *                 *
 *                 *
 *                **************************   ********************************
 *                *FOR I = 0, 1, HEAD'NO $ *---*HEADER($I$) = CNVERT(HEADER($I*
 *                **************************   *$)) $                          *
 *                 *                           ********************************
 *                 *
 *                *******************************
 *                *BYTE($PAGE'WIDTH - 10,       *
 *                * 5$)(HEADER($0$)) = 5H(PAGE )*
 *                * $                           *
 *                *NAME'BYTE = MARGIN +         *
 *                * LENGTH(HEADER($NAME'INDEX$))*
 *                * + 1 $                       *
 *                *******************************
 *
****************   ***************************   ********************************
*IF TITLE'SW $ *---*FOR I = 0, 1, TITLE'NO $ *---*TITLE($I$) = CNVERT(TITLE($I$)*
****************   ***************************   *) $                           *
 *                                               ********************************
 *
*********************************
*FOR I = 0, 1, OUT'BUF'SIZE - 1 *   **************************
* $                             *---*OUT'LINE($I$) = 1H( ) $ *
*                               *   **************************
*********************************
 *
 *
***********
*RETURN $ *
```

```
***********
```

```
DESIGN DIAGRAM OF PH2ERR

****************************
*PROC PH2ERR(ERROR"MESS) $  *
****************************
*
*
*
* "" THE PROCEDURE PH2ERR ISSUES
    ERROR MESSAGES AND STOPS
    EXECUTION OF THE DDG. ""
*
*
******************************
*ITEM ERROR"MESS CHARACTER $ *
*"" THE ERROR MESSAGE""      *
*OUT(ERROR"MESS, MESS"SW) $  *
******************************
*
*
***************
*IF DEBUG27 $ *
*""18""       *----+
*             *    : ""WRITE BLOCKS TO DISK ""
***************    :
*                  :
*                  *********    *************    ************************
*                  *IFEITH  *-+-*NEW"FILE3 $ *---*MESSAGE = 6H(FILE31) $ *
*                  *********  : *************    ************************
*                  :          :
*                  :          :  **********     ************************
*                  :          +-*ORIF 1 $  *---*MESSAGE = 6H(FILE32) $ *
*                  :             **********     ************************
*                  :
*                  *************************
*                  *OUT(MESSAGE, MESS"SW) $ *
*                  *TRANSFER"WRITE3 $       *
*                  *************************
*                  :
*                  :
*                  *********    *************    ************************
*                  *IFEITH  *-+-*NEW"FILE4 $ *---*MESSAGE = 6H(FILE41) $ *
*                  *********  : *************    ************************
*                  :          :
*                  :          :  **********     ************************
*                  :          +-*ORIF 1 $  *---*MESSAGE = 6H(FILE42) $ *
*                  :             **********     ************************
*                  :
*                  *************************
*                  *OUT(MESSAGE, MESS"SW) $ *
*                  *TRANSFER"WRITE4 $       *
*                  *************************
*
*
*********
*STOP $ *
*********
```

DESIGN DIAGRAM OF PLACE

```
**************************
*PROC PLACE(NEW'BOX) $  *
**************************
 *
 *
 *
 * '' THE PROCEDURE PLACE  PLACES
 *  THE CURRENT CODE BLOCK ON THE
 *  DESIGN DIAGRAM. IT ALSO
 *  PERFORMS STUMP DETECTION. ''
 *
 *
**********************************
*ITEM BOTTOM INTEGER $           *
*''PAGE ON WHICH THE BOX BOTTOM*
* APPEARS''                      *
*ITEM ENTRANCE INTEGER $         *
*''VERTICAL OR HORIZONTAL ENTRY*
* INTO BLOCK''                   *
*ITEM FATHER'BOTTOM INTEGER $    *
*''BOTTOM LINE OF FATHER CODE    *
* BLOCK''                        *
*ITEM FATHER'LEFT INTEGER $      *
*''FATHER'S STARTING COLUMN''    *
*ITEM FATHER'RIGHT INTEGER $     *
*'' FATHER'S ENDING COLUMN''     *
*ITEM FATHER INTEGER $           *
*''FATHER'S FILE3 RECORD         *
* NUMBER''                       *
*ITEM FATHER'TYPE INTEGER $      *
*''FATHER CODE BLOCK'S TYPE''    *
*ITEM HORIZ INTEGER P 0 $        *
*'' CONSTANT INDICATING HORIZ    *
* ENTRY''                        *
*ITEM MID INTEGER $              *
*'' PAGE ON WHICH BLOCK'S        *
* MIDPOINT APPEARS''             *
*ITEM NEW'BOX INTEGER $          *
*''NEW F3 REC DESCRIBING LATEST*
* BOX ''                         *
*ITEM PAGE'SPANS INTEGER $       *
*''NUMBER OF PAGES SPANNED BY    *
* BLOCK''                        *
*ITEM REM1 INTEGER $             *
*'' PAGE DISPLACEMENT''          *
*ITEM REM2 INTEGER $             *
*''PAGE DISPLACEMENT''           *
*ITEM REM3 INTEGER $             *
*'' PAGE DISPLACEMENT''          *
*ITEM STMT'TYPE INTEGER $        *
*''TYPE OF CURRENT FILE3 REC''   *
*ITEM STUMP'REF'BOTTOM INTEGER   *
* $                              *
*''BOTTOM OF STUMP REF           *
* DISPLAY''                      *
*ITEM TOP INTEGER $              *
```

```
DESIGN DIAGRAM OF PLACE

*''PAGE ON WHICH TOP OF BLOCK  *
* APPEARS''                    *
*ITEM VERT INTEGER P 1 $       *
*''CONSTANT INDICATING VERTICAL*
* ENTRY''                      *
********************************
 *
 *
****************
*IF DEBUG20 $  *    ********************************
*''7''          *---*MESSAGE = 6H(PLACE ) $        *
*              *    *T'MESS = IFORMAT(NEW'BOX) $   *
****************    *MESSAGE = CAT(MESSAGE, T'MESS)*
 *                  * $                            *
 *                  *OUT(MESSAGE, MESS'SH) $       *
 *                  ********************************
 *
********************************
*STUMP'FOUND = FALSE $         *
*'' ASSUME IT'S NOT A STUMP '' *
*STMT'TYPE =(/ STMT'UNIT($ACCES*
*S3(NEW'BOX)$) /) $            *
********************************
 *
 *
 *
 * '' DETERMINE THE ACTUAL BOX
 *   WIDTH ''
 *
 *
********************************
*WIDTH = BLOCK'WIDTH($ACCESS3(N*
*EW'BOX)$) $                   *
********************************
 *
 *
**********   ****************************    ******************************
*IFEITH  *-+-*STMT'TYPE EQ COMMENT'2 $  *---*WIDTH = WIDTH + 3 $         *
********** * ****************************    *'' ADD COLUMNS FOR IMAGINARY*
 *         *                                * BOX''                      *
 *         *                                ******************************
 *         *
 *         * *********************************
 *         +-*ORIF STMT'TYPE GQ CONTROL'1 $  *----+
 *         * *********************************    * '' NEED A POINTED BOX ''
 *         *                                      *
 *         *                                      *
 *         *                                     ********************************
 *         *                                     *WIDTH = WIDTH + 2 +           *
 *         *                                     * LINES($ACCESS3(NEW'BOX)$) / 2*
 *         *                                     * $                            *
 *         *                                     *WIDTH = MIN(WIDTH, MAX'WIDTH) *
 *         *                                     * $                            *
 *         *                                     *''TRUNCATE IF NECESSARY ''    *
 *         *                                     ********************************
 *         *                                      *
 *         *                                      *
 .         .                                     *******************************
 .         .                                     *IF STMT'TYPE EQ CONTROL'3 $  *----+
 .         .                                     *******************************    . '' ADD ROOM FOR 30X TAIL ''
 .         .                                                                        .
 .         .                                                                        .
 .         .                                                                       *****************************
 .         .                                                                       *WIDTH = WIDTH + BOX'TAIL $ *
 .         .                                                                       *****************************
 .         .
 .         . ************
 .         +-*ORIF 1 $  *----+
 .           ************    . '' A RECTANGULAR BOX ''
 .                           .
 .                           .
 .                          **********************
 .                          *WIDTH = WIDTH + 2 $ *
 .                          **********************
 .
**********   ****************************
*IFEITH  *-+-*STMT'TYPE NQ CONTROL'1 $  *----+
********** . ****************************    . '' BOX HAS A FATHER -- IT'S
 .         .                                 .  NOT A PROC,CLOSE OR PROGRAM
 .         .                                 .  HEAD''
 .         .                                 .
 .         .                                 .
 .         .                                **********   ********************************
 .         .                                *IFEITH  *-+-*BACK'H($ACCESS3(NEW'BOX)$) NQ *   ********************************
 .         .                                ********** . * 0 $                          *---*FATHER = BACK'H($ACCESS3(NEW'B*
 .         .                                 .         . *                              *   *OX)$) $                       *
 .         .                                 .         . ********************************   *FATHER'RIGHT = STOP'COL($ACCES*
 .         .                                 .         .                                    *S3((/ FATHER /))$) $          *
 .         .                                 .         .                                    *ENTRANCE = HORIZ $            *
 .         .                                 .         .                                    ********************************
 .         .                                 .         .
 .         .                                 .         . *********************************
 .         .                                 .         +-*ORIF BACK'V($ACCESS3(NEW'BOX)$ *   ********************************
 .         .                                 .         . *) NQ 0 $                       *---*FATHER = BACK'V($ACCESS3(NEW'B*
 .         .                                 .         . *                               *   *OX)$) $                       *
 .         .                                 .         . *********************************   *FATHER'BOTTOM =               *
 .         .                                 .         .                                     * STOP'LINE($ACCESS3((/ FATHER *
 .         .                                 .         .                                     * /))$) $                      *
 .         .                                 .         .                                     *FATHER'LEFT = START'COL($ACCES*
 .         .                                 .         .                                     *S3((/ FATHER /))$) $          *
 .         .                                 .         .                                     *ENTRANCE = VERT $             *
 .         .                                 .         .                                     ********************************
 .         .                                 .         .
 .         .                                 .         . ************   *******************************
 .         .                                 .         +-*ORIF 1 $  *---*MESSAGE = 14H(BACK PTR ERROR)*
 .         .                                 .           ************   * $                           *
 .         .                                 .                          *PH2ERR(MESSAGE) $            *
 .         .                                 .                          *******************************
 .         .                                 .
 .         .                                **********   *****************
 .         .                                *IFEITH  *-+-*FATHER GR 0 $  *----+
 .         .                                ********** . *****************    . '' WE ARE NOT BEGINNING A
 .         .                                           .                      .  STUMP ''
```

```
DESIGN DIAGRAM OF PLACE
                                                                              .
 .         .                                           .                      .
 .         .                                           .                     ********************************
 .         .                                           .                     *FATHER'TYPE =(/               *
 .         .                                           .                     * STMT'UNIT($ACCESS3(FATHER)$)*
 .         .                                           .                     * /) $                         *
 .         .                                           .                     ********************************
 .         .                                           .
 .         .                                           .
 .         .                                           .                     **********   ***********************  ******
 .         .                                           .                     *IFEITH  *-+-*ENTRANCE EQ HORIZ $  *--* 124*
 .         .                                           .                     ********** . ***********************  ******
 .         .                                           .                                .
 .         .                                           .                                .
 .         .                                           .                                . ***************************  ******
 .         .                                           .                                +-*ORIF ENTRANCE EQ VERT $  *--* 125*
 .         .                                           .                                  ***************************  ******
 .         .                                           .
 .         .                                             **********************
 .         .                                           +-*ORIF FATHER LS 0 $  *----+
 .         .                                             **********************    .  " A STUMP ROOT -- NEW PAGE "
 .         .                                                                       .
 .         .                                                                       .  " MAKE ROOM FOR STUMP DISPLAY
 .         .                                                                       .   "
 .         .                                                                       .
 .         .                                                                       .
 .         .                                                                      **************************  ******
 .         .                                                                      *IF ENTRANCE EQ VERT $   *--* 126*
 .         .                                                                      **************************  ******
 .         .                                                                       .
 .         .                                                                       .
 .         .                                                                      ********************************
 .         .                                                                      *PUSH'LAYOUT'INFO $            *
 .         .                                                                      *INSERT(STUMP'ROOT, NEW'BOX) $ *
 .         .                                                                      *START'LINE($ACCESS3(NEW'BOX)$)*
 .         .                                                                      * = HEADROOM - 1 +             *
 .         .                                                                      * DISPLAY'LINES + V'SPACE $    *
 .         .                                                                      *START'COL($ACCESS3(NEW'BOX)$) *
 .         .                                                                      * = MARGIN $                   *
 .         .                                                                      ********************************
 .         .
 .         .  *********************************
 .         +-*ORIF STMT'TYPE EQ CONTROL'1 $  *----+
 .           *********************************    .  " A PROGRAM,PROC OR CLOSE
 .                                                .  HEAD -- NEW PAGE "
 .                                                .
 .                                                .
 .                                               ********************************
 .                                               *START'LINE($ACCESS3(NEW'BOX)$)*
 .                                               * = HEADROOM - 1 $             *
 .                                               *START'COL($ACCESS3(NEW'BOX)$) *
 .                                               * = MARGIN $                   *
 .                                               ********************************
 .
 .
 .  " IS NEW BOX A STUMP  "
 .
```

```
DESIGN DIAGRAM OF PLACE

*********************************
*STOP'COL($ACCESS3(NEW'BOX)$) =*
* START'COL($ACCESS3(NEW'BOX)$)*
* + WIDTH - 1 $                 *
*WRITE3 = TRUE $                *
*********************************
 .
 .
*********************************
*IF STOP'COL($ACCESS3(NEW'BOX)$ *
*) GR PAGE'WIDTH $               *---+-+
*                               *     . **IT'S A STUMP **
*********************************     .
 .                                    .
 .                                   ***********************
 .                                   *STUMP'FOUND = TRJE $ *
 .                                   *RETURN $             *
 .                                   ***********************
 .
*********************************
*STOP'LINE($ACCESS3(NEW'BOX)$) *
* = START'LINE($ACCESS3(NEW'BOX*
*)$) + LINES($ACCESS3(NEW'BOX)$*
*) - 1 $                        *
*********************************
 .
 .
*********    ************
*IFEITH  *-+-*HEADING $  *----+
*********  . ************     . **DOES NEW BOX SPAN A PAGE
 .         .                  .   HEADING  **
 .         .                  .
 .         .                  .
 .         .                 ********************************
 .         .                 *REMQUO(START'LINE($ACCESS3(NEW*
 .         .                 **BOX)$), PAGE'LNGTH = TOP,    *
 .         .                 * REM1) $                      *
 .         .                 *REMQUO(STOP'LINE($ACCESS3(NEW**
 .         .                 *BOX)$), PAGE'LNGTH = BOTTOM,  *
 .         .                 * REM2) $                      *
 .         .                 ********************************
 .         .                  .
 .         .                  .
 .         .                 ************************    *********************************
 .         .                 *IF REM1 LQ HEADROOM $  *---*START'LINE($ACCESS3(NEW'BOX)$)*
 .         .                 ************************    * = START'LINE($ACCESS3(NEW'BOX*
 .         .                  .                          *)$) + HEADROOM - REM1 $       *
 .         .                  .                          *STOP'LINE($ACCESS3(NEW'BOX)$) *
 .         .                  .                          * = STOP'LINE($ACCESS3(NEW'BOX)*
 .         .                  .                          *$) + HEADROOM - REM1 $        *
 .         .                  .                          *********************************
 .         .                  .
 .         .                 *******************************
 .         .                 *PAGE'SPANS = BOTTOM - TOP $ *
 .         .                 *******************************
 .         .
 .         .                  **********   ********************
 .         .                  *IFEITH  *-+-*PAGE'SPANS GR 0 $ *----+
 .         .                  ********** . ********************    . ''IT SPANS A HEADING ''
 .         .                             .                         .
 .         .                             .                         .
 .         .                             .                         ****************
 .         .                             .                         *IF DEBUG20 $  *   **************************
 .         .                             .                         *''5''         *---*MESSAGE = 5H(SPANS) $   *
 .         .                             .                         *              *   *OUT(MESSAGE, MESS'SW) $ *
 .         .                             .                         ****************   **************************
 .         .                             .
 .         .                             .
 .         .                             .
 .         .                             .
 .         .                             .
 .         .                             .
 .         .                             .
 .         .                             . ************
 .         .                             +-*ORIF 1 $  *----+
 .         .                               ************    . ''DOESN'T SPAN HEADING ''
 .         .                                               .
 .         .                                               .
 .         .                                              ****************
 .         .                                              *IF DEBUG20 $  *   **************************
 .         .                                              *''5''         *---*MESSAGE = 7H(NO SPAN) $ *
 .         .                                              *              *   *OUT(MESSAGE, MESS'SW) $ *
 .         .                                              ****************   **************************
 .         .                                               .
 .         .                                               .
 .         .                                              *******************************   ********************************
 .         .                                              *IF STMT'TYPE GQ CONTROL'1 $  *---*MIDPOINT = (START'LINE($ACCESS*
 .         .                                              *******************************   *3(NEW'BOX)$) + STOP'LINE($ACCE*
 .         .                                                                                *SS3(NEW'BOX)$)) / 2 $         *
 .         .                                                                                ********************************
 .         .
 .         . *********************************
 .         +-*ORIF NOT HEADING AND STMT'TYPE *   ********************************
 .           * GQ CONTROL'1 $                *---*MIDPOINT = (START'LINE($ACCESS*
 .           *                               *   *3(NEW'BOX)$) + STOP'LINE($ACCE*
 .           *********************************   *SS3(NEW'BOX)$)) / 2 $         *
 .                                               ********************************
 .
 .
 . ''UPDATE LAST'LINE ''
 .
 .
********************************
*LAST'LINE = MAX(LAST'LINE,    *
* STOP'LINE($ACCESS3(NEW'BOX)$)*
*) $                           *
********************************
 .
 .
*******************************   *****************************
*IF STMT'TYPE GQ CONTROL'1 $  *---*MIDPT($ACCESS3(NEW'BOX)$) =*
*******************************   * MIDPOINT $                *
```

```
                                   *****************************

***********
*RETURN $ *
***********
```

```
DESIGN DIAGRAM OF PLACE

****************
* 124 FROM 120 *
****************
 *
 *
 *
**********   *****************************   ********************************
*IFEITH  *-+-*STMT'TYPE EQ COMMENT'2 $   *---*START'LINE($ACCESS3(NEW'BOX)$)*
********** * *****************************   * = MIDPOINT $                  *
 *         *                                 ********************************
 *         *
 *         * ************    ********************************
 *         +-*ORIF 1 $  *---*START'LINE($ACCESS3(NEW'BOX)$)*
 *           ************   * = MIDPOINT - 1 $             *
 *                          ********************************
 *
**********   *****************************   *******************************
*IFEITH  *-+-*STMT'TYPE EQ CONTROL'3 $   *---*START'COL($ACCESS3(NEW'BOX)$)*
********** * *****************************   * = FATHER'RIGHT + H'SPACE'2 $*
           *                                 *******************************
           *
           * ************    *******************************
           +-*ORIF 1 $  *---*START'COL($ACCESS3(NEW'BOX)$)*
             ************   * = FATHER'RIGHT + H'SPACE'1 $*
                            *******************************
```

```
DESIGN DIAGRAM OF PLACE

****************
* 125 FROM 120 *
****************
 *
 *
 *
**********   *******************************
*IFEITH  *-+-*FATHER'TYPE EQ CONTROL'3 AND *   *******************************
********** * * STMT'TYPE NQ CONTROL'3 $    *---*START'COL($ACCESS3(NEW'BOX)$)*
 *         * *                             *   * = FATHER'LEFT - 1 $          *
 *         * *******************************   *******************************
 *         *
 *         *
 *         * ********************************   *******************************
 *         +-*ORIF FATHER'TYPE NQ CONTROL'3 *   *******************************
 *         * * AND STMT'TYPE EQ CONTROL'3 $ *---*START'COL($ACCESS3(NEW'BOX)$)*
 *         * *                              *   * = FATHER'LEFT + 1 $          *
 *         * ********************************   *******************************
 *         *
 *         *
 *         * ***********    *******************************
 *         +-*ORIF 1 $  *---*START'COL($ACCESS3(NEW'BOX)$)*
 *            ***********   * = FATHER'LEFT $             *
 *                          *******************************
 *
**********   *****************************    ********************************
*IFEITH  *-+-*FATHER'TYPE EQ COMMENT'2 $  *---*START'LINE($ACCESS3(NEW'BOX)$)*
********** * *****************************    * = MAX(LAST'LINE + 2,         *
           *                                  * FATHER'BOTTOM + 1) $         *
           *                                  ********************************
           * ***********    ********************************
           +-*ORIF 1 $  *---*START'LINE($ACCESS3(NEW'BOX)$)*
              ***********   * = MAX(LAST'LINE + 2,         *
                            * FATHER'BOTTOM + V'SPACE) $   *
                            ********************************
```

```
C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER
DESIGN DIAGRAM OF PLACE

****************
* 126 FROM 120 *
****************
 *
 *
 *
*******************************
*STUMP'REF'BOTTOM =           *
* STOP'LINE($ACCESS3((/ FATHER*
* /))$) + DISPLAY'LINES +     *
* V'SPACE $                   *
*REMQUO(STUMP'REF'BOTTOM,     *
* PAGE'LNGTH = BOTTOM, REM1) $*
*******************************
 *
 *
*****************************
*IF REM1 - DISPLAY'LINES LS *    ********************************
* HEADROOM $                 *---*STUMP'REF'BOTTON =            *
*                           *    * STUMP'REF'BOTTOM + HEADROOM -*
*****************************    * REM1 $                       *
 *                               ********************************
 *
*****************************
*LAST'LINE = MAX(LAST'LINE,*
* STUMP'REF'BOTTOM) $       *
*****************************
```

```
C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER
DESIGN DIAGRAM OF PLACE                                                                                           PAGE 127

****************
* 127 FROM 0   *
****************
 .
 .
 .
*********   **************************
*IFEITH  *-+-*STMT'TYPE GQ CONTROL'1 $  *----+
*********  . **************************      . ''TRY NOT TO LET A CONTROL BOX
           .                                 .  SPAN A HEADING ''
           .                                 .
           .                               *********    ****************************
           .                               *IFEITH  *-+-*LINES($ACCESS3(NEW'BOX)$) GR *
           .                               *********  . * PAGE'LNGTH - HEADROOM $ LIN  *----+
           .                                          . *                              *    . ''IT CAN'T BE AVOIDED ''
           .                                          . ****************************        .
           .                                          .                                    ******************************
           .                                          .                                    *REMQUO(MIDPOINT, PAGE'LNGTH =*
           .                                          .                                    * MID, REM3) $               *
           .                                          .                                    ******************************
           .                                          .                                     .
           .                                          .                                     .
           .                                          .                                    ******
           .                                          .                                    * 129*
           .                                          .                                    ******
           .                                          . ************
           .                                          +-*ORIF 1 $  *----+
           .                                            ************    . '' IT CAN BE AVOIDED-- MOVE
           .                                                            .  BOX TO NEXT PAGE ''
           .                                                            .
           .                                                           ******************************
           .                                                           *START'LINE($ACCESS3(NEW'BOX)$)*
           .                                                           * = STOP'LINE($ACCESS3(NEW'BOX)*
           .                                                           *$) - REM2 + HEADROOM $       *
           .                                                           *STOP'LINE($ACCESS3(NEW'BOX)$) *
           .                                                           * = START'LINE($ACCESS3(NEW'BOX*
           .                                                           *)$) + LINES($ACCESS3(NEW'BOX)$*
           .                                                           *) - 1 $                      *
           .                                                           *MIDPOINT = (START'LINE($ACCESS*
           .                                                           *3(NEW'BOX)$) + STOP'LINE($ACCE*
           .                                                           *SS3(NEW'BOX)$)) / 2 $        *
           .                                                           ******************************
           .                                                            .
           .                                                            .
           .                                                           ******
           .                                                           * 130*
           .                                                           ******
         ************
         +-*ORIF 1 $  *----+
         ************      . '' NOT A CONTROL BOX ''
         *
         *
********************************
*STOP'LINE($ACCESS3(NEW'BOX)$) *
* = STOP'LINE($ACCESS3(NEW'BOX)*
*$) + (HEADROOM - 1)           *
* PAGE'SPANS $                 *
********************************
```

```
DESIGN DIAGRAM OF PLACE

****************
* 129 FROM 127 *
****************
 *
 *
 *
*******************
*IF TOP NQ MID $  *----+
*******************    *  **ADJUST MIDPOINT **
 *                     *
 *                     *
 *                    ********************************
 *                    *MIDPOINT = MIDPOINT + HEADROOM*
 *                    * * (MID - TOP) $              *
 *                    ********************************
 *
********************************
*STOP'LINE($ACCESS3(NEW'BOX)$) *
* = STOP'LINE($ACCESS3(NEW'BOX)*
*$) + (HEADROOM - 1) *         *
* (PAGE'SPANS) $               *
********************************
```

```
DESIGN DIAGRAM OF PLACE

****************
* 130 FROM 127 *
****************
 *
 *
 *
**************************
*IF ENTRANCE EQ HORIZ $  *-----+
**************************     * ** MOVE BOX BACK 1 COLUMN **
                               *
                               *
                              *********************************
                              *START'COL($ACCESS3(NEW'BOX)$) *
                              * = START'COL($ACCESS3(NEW'BOX)*
                              *$) - 1 $                       *
                              *STOP'COL($ACCESSS3(NEW'BOX)$) =*
                              * STOP'COL($ACCESS3(NEW'BOX)$) *
                              * - 1 $                         *
                              *********************************
```

```
C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER                                                    PAGE 131
DESIGN DIAGRAM OF POP'LAYOUT'INFO

**************************
*PROC POP'LAYOUT'INFO $  *
**************************
 .
 .
 .
 . '' THE PROCEDURE
 .  POP'LAYOUT'INFO POPS THE TOP
 .  ELEMENT OF LAYOUT'STACK INTO
 .  LAST'LINE. ''
 .
 .
**********   ****************************   *****************************
*IFEITH  *-+-*LAYOUT'STACK'TOP EQ D $  *---*MESSAGE = 28H(PROC OR STUMP*
**********  . ****************************   * HANDLING ERROR) $          *
            .                                *PH2ERR(MESSAGE) $           *
            .                                *****************************
            .
            . ************   ****************   *******************************
            +-*ORIF 1 $  *---*IF HEADING $  *---*PAGE'REF($CUR'GROUP$) =       *
              ************   ****************   * LAST'LINE / PAGE'LNGTH + 1 $*
                              .                 *******************************
                              .
                             ********************************
                             *LAST'LINE = LAYOUT'STACK($LAYO*
                             *UT'STACK'TOP$) $              *
                             *CUR'GROUP = GROUP'STACK($LAYOU*
                             *T'STACK'TOP$) $               *
                             *LAYOUT'STACK'TOP =            *
                             * LAYOUT'STACK'TOP - 1 $       *
                             *RETURN $                      *
                             ********************************
```

```
C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER                                                 PAGE 132
DESIGN DIAGRAM OF POP'PROC'STACK


********************************
*PROC POP'PROC'STACK(= POP'REC) *
* $                              *
*                               *
********************************
 *
 *
 *
 * '' THE PROCEDURE POP'REC POPS
 *   THE TOP ELEMENT OF PROC'STACK
 *   ONTO POP'REC. ''
 *
 *
************************
*ITEM POP'REC INTEGER $ *
*''OUTPUT PARAMETER''   *
************************
 *
 *
***************
*IF DEBUG21 $  *    ******************************
*''7''         *----*MESSAGE = 8H(POP OFF ) $      *
*              *    *T'MESS = IFORMAT(PROC'STACK($P*
***************     *ROC'STACK'TOPS)) $           *
 *                  *MESSAGE = CAT(MESSAGE, T'MESS)*
 *                  * $                           *
 *                  *OUT(MESSAGE, MESS'SW) $      *
 *                  ******************************
 *
*************************    ******************************
*IF PROC'STACK'TOP LS 1 $ *---*MESSAGE = 19H(PROC HANDLING*
*************************    * ERROR) $                  *
 *                           *PH2ERR(MESSAGE) $          *
 *                           ******************************
 *
******************************
*POP'REC = PROC'STACK($PROC'STA*
*CK'TOPS) $                   *
*PROC'STACK'TOP =             *
* PROC'STACK'TOP - 1 $ PROC'STA*
******************************
 *
 *
*************************    **********************
*IF PROC'STACK'TOP EQ 1 $ *---*PROC'FLAG = FALSE $ *
*************************    **********************
 *
 *
***********
*RETURN $ *
***********
```

```
***************************
*PROC PUSH*LAYOUT*INFO $  *
***************************
 *
 *
 *
 * ** THE PROCEDURE
 *  PUSH*LAYOUT*INFJ PUSHES THE
 *  VALUE OF LAST*LINE ONTO THE
 *  TOP OF LAYOUT*STACK. **
 *
 *
**********   ***********************
*IFEITH  *-+-*LAYOUT*STACK*TOP GR *     *******************************
**********  * * LAYOUT*STACK*MAX $  *---*MESSAGE = 21H(LAYOUT STACK*
            * *                    *    * OVERFLOW) $                 *
            * **********************    *PH2ERR(MESSAGE) $            *
            *                           *******************************
            *
            * ************    ************************************
            +-*ORIF 1 $  *---*LAYOUT*STACK*TOP =                *
              ************   * LAYOUT*STACK*TOP + 1 $           *
                             *GROUP*STACK($LAYOUT*STACK*TOP$*
                             *) = CUR*GROUP $                  *
                             *LAYOUT*STACK($LAYOUT*STACK*TOP*
                             *$) = LAST*LINE $                 *
                             *LAST*LINE = 0 $                  *
                             *RETURN $                         *
                             ************************************
```

```
*********************************
*PROC PUSH*PROC*STACK(PUSH*REC) *
* $                               *
*                               *
*********************************
 *
 *
 *
 * ** THE PROCEDURE
 *   PUSH*PROC*STACK PUSHES THE
 *   NUMBER OF THE CURRENT FILE 3
 *   RECORD ONTO THE PROC*STACK.
 *   **
 *
 *
*********************************
*ITEM PUSH*REC INTEGER $        *
*** THE NUMBER OF THE RECORD** *
*********************************
 *
 *
***************
*IF DEBUG21 $  *     *********************************
***7**          *-:-*MESSAGE = 5H(PUSH ) $         *
*              *     *T*MESS = IFORMAT(PUSH*REC) $  *
***************      *MESSAGE = CAT(MESSAGE, T*MESS)*
 *                   * $                            *
 *                   *OUT(MESSAGE, MESS*SW) $       *
 *                   *********************************
 *
 *
 * ** THIS PROC IS NOT NECESSARY
 *   FOR JOVIAL, BUT LANGUAGES **
 * ** SUPPORTING NESTED PROCS
 *   WOULD REQUIRE IT. **
 *
 *
**********    ********************
*IFEITH  *-+-*PROC*STACK*TOP EQ *     ***************************
**********  * * PROC*STACK*MAX $  *---*MESSAGE = 19H(PROC STACK*
            * *                 *     * OVERFLOW) $             *
            * ********************    *PH2ERR(MESSAGE) $        *
            *                         ***************************
            *
            * ***********     *********************************
            +-*ORIF 1 $  *---*PROC*STACK*TOP =                *
              ***********     * PROC*STACK*TOP + 1 $          *
                              *PROC*STACK($PROC*STACK*TOP$) =*
                              * PUSH*REC $                    *
                              *********************************
                               *
                               *
                              ****************************     *********************
                              *IF PROC*STACK*TOP GR 1 $  *---*PROC*FLAG = TRUE $ *
                              ****************************     *********************
```

```
***********
*RETURN $ *
***********
```

```
DESIGN DIAGRAM OF RESOLVE'STUMP

*******************************
*PROC RESOLVE'STUMP(STUMP'REC) *
* $                             *
*                               *
*******************************
 *
 *
 *
 * '' THE PROCEDURE
 *   RESOLVE'STUMP  FINDS THE ROOT
 *   OF A STUMP AND INVOKES PLACE
 *   FOR EACH RECORD CURRENTLY
 *   HANGING OFF THE STUMP ROOT.
 *   ''
 *
 *
********************************
*ITEM DISPLAY'ROOM B $         *
*''ENOUGH ROOM FOR STUMP REF   *
* DISPLAY ''                   *
*ITEM DONE B $                 *
*''COMPLETION FLAG''           *
*ITEM FATHER INTEGER $         *
*''RECORD NUMBER OF FATHER''   *
*ITEM HORIZ INTEGER P 0 $      *
*'' CONSTANT INDICATING HORIZ  *
* ENTRY''                      *
*ITEM INDEX INTEGER $          *
*''USED AS AN INDEX INTO THE   *
* GROUP TABLE''                *
*ITEM INIT'STUMP INTEGER $     *
*''RECORD NUMBER OF THE REC    *
* CAUSING STUMP''              *
*ITEM LAST'INDEX INTEGER $     *
*''STORES  PREVIOUS INDEX INTO *
* GROUP''                      *
*ITEM OLD'INDEX INTEGER $      *
*''PREVIOUS GROUP-INDEX''      *
*ITEM STUMP'REC INTEGER $      *
*'' CURRENT FILE3 RECORD''     *
*ITEM SUB'STUMP INTEGER $      *
*''RECORD NUMBER OF SUB STUMP''*
*                              *
*ITEM TYPE INTEGER P 0 $       *
*'' HORIZ-VERT FLAG''          *
*ITEM VERT INTEGER P 1 $       *
*'' CONSTANT INDICATING VERT   *
* ENTRY''                      *
********************************
 *
 *
***************
*IF DEBUG26 $ *    ****************************
*''7''        *---*MESSAGE = 16H(STUMP CAUSE REC *
*             *    * ) $                        *
***************    *T'MESS = IFORMAT(STUMP'REC) $ *
```

```
C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER
DESIGN DIAGRAM OF RESOLVE'STUMP
                    *MESSAGE = CAT(MESSAGE, T'MESS)*
 *                  * $                            *
 *                  *OUT(MESSAGE, MESS'SW) $       *
 *                  ********************************
 *
 *
 * ** MAKE SURE THAT STUMP IS
 *   LEGAL AND THAT ITS FATHER HAS
 *   ROOM **
 * ** FOR A STUMP PAGE REFERENCE
 *   DISPLAY. **
 *
 *
***************************
*STUMP'FOUND = FALSE $    *
*** RESET **              *
*DONE = FALSE $           *
*INIT'STUMP = STUMP'REC $ *
*** SAVE NO. OF REC CAUSING*
* STUMP**                 *
***************************
 *
 *
*****************
*DO UNTIL [DONE]  *----+
*****************      * **TEST LEGALITY **
 *                     *
 *                     *
 *                    ********************************
 *                    *DO WHILE [NOT (LEGAL'STUMP((/  *    ********************************
 *                    * STMT'UNIT($ACCESS3(STUMP'REC)  *---*STUMP'REC = MAX(BACK'H($ACCESS*
 *                    *$) /))))]                      *    *3(STUMP'REC)$),               *
 *                    ********************************     * BACK'V($ACCESS3(STUMP'REC)$))*
 *                     *                                   * $                            *
 *                     *                                   ********************************
 *                     *                                    *
 *                     *                                    *
 *                     *                                   **********************     ******************************
 *                     *                                   *IF STUMP'REC LQ 0 $  *---*MESSAGE = 20H(STUMP HANDLING*
 *                     *                                   **********************    * ERROR) $                     *
 *                     *                                                             *PH2ERR(MESSAGE) $             *
 *                     *                                                             ******************************
 *                     *
 *                    ***************
 *                    *DONE = TRUE $ *
 *                    ***************
 *                     *
 *                     *
 *                     *
 *                     * ** ROOM FOR DISPLAY BOX  **
 *                     *
 *                     *
 *                    *************************
 *                    *DISPLAY'ROOM = FALSE $  *
 *                    *** ASSUME THERE ISN'T ** *
 *                    *************************
 *
```

```
*                     *******************************
*                     *DO UNTIL [DISPLAY'ROOM OR NOT *   *********   *******************************
*                     * (DONE)]                      *---*IFEITH *-+-*BACK'H($ACCESS3(STUMP'REC)$) *     ******
*                     *                              *   ********* * * GR 0 $                         *--* 144*
*                     *******************************              * *                             *     ******
*                                                                  * *******************************
*                                                                  *
*                                                                  *
*                                                                  * *********************************
*                                                                  +-*ORIF BACK'V($ACCESS3(STUMP'REC *    ******
*                                                                  * *)$) GR 0 $                       *--* 145*
*                                                                  * *                               *    ******
*                                                                  * *********************************
*                                                                  *
*                                                                  *
*                                                                  * ************   ******************************
*                                                                  +-*ORIF 1 $  *---*MESSAGE = 22H(STUMP HANDLING*
*                                                                    ************   * ERROR 2) $                 *
*                                                                                   *PH2ERR(MESSAGE) $           *
*                                                                                   ******************************
*
*
* ** RESET LAST'LINE **
*
*
********************************
*LAST'LINE = STOP'LINE($ACCESS3*
*(FATHER)$) $                  *
********************************
*
*
*
* **NEGATE POINTERS **
*
*
*********    ******************   ********************************
*IFEITH  *-+-*TYPE EQ.VERT $  *---*BACK'V($ACCESS3(STUMP'REC)$) =*
*********  * ******************   * - FATHER $                   *
*          *                      *WRITE3 = TRUE $               *
*          *                      *V'PTR($ACCESS3(FATHER)$) = -  *
*          *                      * STUMP'REC $                  *
*          *                      *WRITE3 = TRUE $               *
*          *                      ********************************
*          *
*          * ***********************    ********************************
*          +-*ORIF TYPE EQ HORIZ $   *---*BACK'H($ACCESS3(STUMP'REC)$) =*
*            ***********************    * - FATHER $                   *
*                                       *WRITE3 = TRUE $               *
*                                       *H'PTR($ACCESS3(FATHER)$) = -  *
*                                       * STUMP'REC $                  *
*                                       *WRITE3 = TRUE $               *
*                                       ********************************
*
*
*
* ** RECONFIGURE STUMP'S PAGE
*  LAYOUT **
```

```
C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER                                                         PAGE 139
DESIGN DIAGRAM OF RESOLVE'STUMP

 *  '' TRAVERSE THE TREE DESCRIBED
 *   BY  FILE 3 AND CALL PLACE ''
 *  '' FOR EACH RECORD ''
 *
 *
********************************
*TRAVERSE'TOP = 0 $            *
*TRAVERSE'STACK($TRAVERSE'TOP$)*
* = 0 $                        *
*SUB'STUMP = 0 $               *
********************************
 *
 *
****************
*IF DEBUG22 $  *    ********************************
*''7''         *---*MESSAGE = 16H(RESOLVING STUMP *
*              *   * ) $                          *
****************   *T'MESS = IFORMAT(STUMP'REC) $ *
 *                 *MESSAGE = CAT(MESSAGE, T'MESS)*
 *                 * $                            *
 *                 *OUT(MESSAGE, MESS'SW) $       *
 *                 ********************************
 *
****************************    ********************************
*DO WHILE [STUMP'REC NQ 0] *---*TRAVERSE'TOP = TRAVERSE'TOP + *
****************************    * 1 $                          *
 *                              *TRAVERSE'STACK($TRAVERSE'TOP$)*
 *                              * = STUMP'REC $                *
 *                              *PLACE(STUMP'REC) $            *
 *                              ********************************
 *                               *
 *                               *
 *                              **********************    **************************
 *                              *IF SUB'STUMP EQ 0 $ *---*OLD'INDEX = CUR'GROUP $ *
 *                              **********************    **************************
 *                               *
 *                               *
 *                              ****************************
 *                              *IF STUMP'FOUND $          *     ********************************
 *                              *''A STUMP AT THIS POINT IS *---*MESSAGE = 27H(STUMP HANDLING  *
 *                              * ILLEGAL ''               *    * ERROR 3 REC ) $              *
 *                              ****************************    *T'MESS = IFORMAT(STUMP'REC) $ *
 *                               *                              *MESSAGE = CAT(MESSAGE, T'MESS)*
 *                               *                              * $                            *
 *                               *                              *PH2ERR(MESSAGE) $             *
 *                               *                              ********************************
 *                               *
 *                               *
 *                               *  '' MAKE CERTAIN THAT  NEITHER
 *                               *   H'PTR NOR V'PTR ''
 *                               *  ''POINTS AT A STUMP -- IF IT
 *                               *   DOES, THE STUMPS MJST BE ''
 *                               *  '' REORDERED WITHIN GROUP''
 *                               *
 *                               *
```

DESIGN DIAGRAM OF RESOLVE'STUMP

```
.                               *********************************
.                               *IF H'PTR($ACCESS3(STUMP'REC)$) *     *********************************
.                               * LS 0 $                          *---*SUB'STUMP =(/ H'PTR($ACCESS3(S*
.                               *                               *     *TUMP'REC)$) /) $               *
.                               *********************************     *LAST'INDEX = 0 $               *
.                                .                                    *INDEX = NEXT($0$) $            *
.                                .                                    *********************************
.                                .                                     .
.                                .                                     .
.                                .                                     .
.                                .                                     . ''FIND THE SUB'STUMP'S ENTRY
.                                .                                     .  IN GROUP''
.                                .                                     .
.                                .                                     .
.                                .                                    *******************************
.                                .                                    *DO WHILE (F3'REC($INDEX$) NQ *    ******
.                                .                                    * SUB'STUMP AND INDEX GR 0)     *--* 146*
.                                .                                    *                             *    ******
.                                .                                    *******************************
.                                .                                     .
.                                .                                     .
.                                .                                    ******************     ***********************************
.                                .                                    *IF INDEX LQ 0 $  *---*MESSAGE = 24H(SUB STUMP SEARCH*
.                                .                                    ******************     * FAILURE) $                      *
.                                .                                     .                     *PH2ERR(MESSAGE) $                *
.                                .                                     .                     ***********************************
.                                .                                     .
.                                .                                     .
.                                .                                     . '' REORDER THE STUMPS''
.                                .                                     .
.                                .                                     .
.                                .                                    *******************************
.                                .                                    *NEXT($LAST'INDEX$) =         *
.                                .                                    * NEXT($INDEX$) $             *
.                                .                                    *NEXT($OLD'INDEX$) = INDEX $ *
.                                .                                    *******************************
.                                .                                     .
.                                .                                     .
.                                .                                    *******************************    ************************
.                                .                                    *IF LAST'PROC EQ OLD'INDEX $  *---*LAST'PROC = INDEX $ *
.                                .                                    *******************************    ************************
.                                .                                     .
.                                .                                     .
.                                .                                    ********************************    *************************
.                                .                                    *IF LAST'STUMP EQ OLD'INDEX $  *---*LAST'STUMP = INDEX $ *
.                                .                                    ********************************    *************************
.                                .                                     .
.                                .                                     .
.                                .                                    ***********************
.                                .                                    *OLD'INDEX = INDEX $ *
.                                .                                    ***********************
.                                .                                     .
.                                .                                     .
.                                .                                     .
.                                .                                     .
.                                .                                     .
.                    .                              ****************
.                    .                              *IF DEBUG26 $  *      ***********************************
.                    .                              *''7''         *---*MESSAGE = 14H(REORDER STUMP )  *
.                    .                              *              *    * $                              *
.                    .                              ****************    *T'MESS = IFORMAT(SUB'STUMP) $    *
.                    .                                                  *MESSAGE = CAT(MESSAGE, T'MESS)*
.                    .                                                  * $                               *
.                    .                                                  *OUT(MESSAGE, MESS'SW) $          *
.                    .                                                  ***********************************
.                    .
.                    **********************************
.                    *IF V'PTR($ACCESS3(STUMP'REC)$) *     ********************************
.                    * LS 0 $                         *---*SUB'STUMP =(/ V'PTR($ACCESS3(S*
.                    *                                *   *TUMP'REC)$) /) $              *
.                    **********************************   *LAST'INDEX = 0 $              *
.                    .                                    *INDEX = NEXT($0$) $           *
.                    .                                    ********************************
.                    .                                    .
.                    .                                    .
.                    .                                    *  ''FIND THE SUB'STUMP'S ENTRY
.                    .                                    *    IN GROUP''
.                    .                                    .
.                    .                                    ********************************
.                    .                                    *DO WHILE [F3'REC($INDEX$) NQ *      ******
.                    .                                    * SUB'STUMP AND INDEX GR 0]    *---* 147*
.                    .                                    *                              *    ******
.                    .                                    ********************************
.                    .                                    .
.                    .                                    *****************     *******************************
.                    .                                    *IF INDEX LQ 0 $ *---*MESSAGE = 24H(SUB STUMP SEARCH*
.                    .                                    *****************     * FAILURE) $                    *
.                    .                                    .                     *PM2ERR(MESSAGE) $              *
.                    .                                    .                     *******************************
.                    .                                    .
.                    .                                    *  '' REORDER THE STUMPS''
.                    .                                    .
.                    .                                    *******************************
.                    .                                    *NEXT($LAST'INDEX$) =        *
.                    .                                    * NEXT($INDEX$) $              *
.                    .                                    *NEXT($OLD'INDEX$) = INDEX $  *
.                    .                                    *******************************
.                    .                                    .
.                    .                                    *******************************     ********************
.                    .                                    *IF LAST'PROC EQ OLD'INDEX $  *---*LAST'PROC = INDEX $ *
.                    .                                    *******************************     ********************
.                    .                                    .
.                    .                                    .
.                    .                                    ********************************     *********************
.                    .                                    *IF LAST'STUMP EQ OLD'INDEX $  *---*LAST'STUMP = INDEX $ *
.                    .                                    ********************************     *********************
```

DESIGN DIAGRAM OF RESOLVE*STUMP

```
.                                .                                   **********************
.                                .                                   *OLD*INDEX = INDEX $ *
.                                .                                   **********************
.                                .                                    .
.                                .                                    .
.                                .                                   ***************
.                                .                                   *IF DEBUG26 $ *     ********************************
.                                .                                   *""7""        *---*MESSAGE = 14H(REORDER STUMP ) *
.                                .                                   *             *     * $                            *
.                                .                                   ***************     *T*MESS = IFORMAT(SUB*STUMP) $ *
.                                .                                                       *MESSAGE = CAT(MESSAGE, T*MESS)*
.                                .                                                       * $                            *
.                                .                                                       *OUT(MESSAGE, MESS*SW) $       *
.                                .                                                       ********************************
.                                .
.                              *********    ********************************
.                              *IFEITH  *-+-*H*PTR($ACCESS3(STUMP*REC)$) GR *     ********************************
.                              *********  * * 0 AND STUMP*REC NQ INIT*STUMP *---*STUMP*REC = H*PTR($ACCESS3(STU*
.                                         * * $                             *     *MP*REC)$) $                   *
.                                         * ********************************      ********************************
.                                         .
.                                         .
.                                         * ********************************
.                                         +-*ORIF V*PTR($ACCESS3(STUMP*REC) *     ********************************
.                                         * *$) GR 0 AND STUMP*REC NQ       *---*STUMP*REC = V*PTR($ACCESS3(STU*
.                                         * * INIT*STJMP $                  *     *MP*REC)$) $                   *
.                                         * ********************************      *TRAVERSE*TOP = TRAVERSE*TOP - *
.                                         .                                       * 1 $                          *
.                                         .                                       ********************************
.                                         .
.                                         * ***********    ***************
.                                         +-*ORIF 1 $  *---*IF DEBUG26 $ *    ********************************
.                                           ***********    *""7""        *---*MESSAGE = 17H(RESOLVE:TREE END*
.                                                          *             *    * ) $                          *
.                                                          ***************    *T*MESS = IFORMAT(STUMP*REC) $ *
.                                                           .                 *MESSAGE = CAT(MESSAGE, T*MESS)*
.                                                           .                 * $                            *
.                                                           .                 *OUT(MESSAGE, MESS*SW) $       *
.                                                           .                 ********************************
.                                                           .
.                                                           .
.                                                           * ** HAVE TRAVERSED RIGHT
.                                                           *  SUB-TREE. LOOK FOR LEFT
.                                                           *  SUB-TREE**
.                                                           * ** TO TRAVERSE **
.                                                           .
.                                                           .
.                                                          ********************************
.                                                          *DO WHILE [(V*PTR($ACCESS3(STUM *
.                                                          *P*REC)$) EQ 0 AND STUMP*REC NQ *    ******
.                                                          * 0) OR STUMP*REC EQ            *--* 148*
.                                                          * INIT*STUMP]                   *    ******
.                                                          *                              *
.                                                          ********************************
.                                                           .
.                                                           .
```

```
                                                                  **********************    ********************************
 .                                                                *IF STUMP'REC NQ 0 $  *---*STUMP'REC = V'PTR($ACCESS3(STU*
 .                                                                **********************    *MP'REC)S) $                   *
 .                                                                                          *TRAVERSE'TOP = TRAVERSE'TOP - *
 .                                                                                          * 1 $                          *
 .                                                                                          ********************************
 .
 .
***********
*RETURN $ *
***********
```

```
DESIGN DIAGRAM OF RESOLVE'STUMP

****************
* 144 FROM 138 *
****************
 *
 *
 *
********************************
*TYPE = HORIZ $                *
*FATHER = BACK'H($ACCESS3(STUMP*
*'REC)$) $                     *
********************************
 *
 *
**********   ********************************
*IFEITH  *-+-*STOP'COL($ACCESS3(FATHER)$) + *    ***********************
********** * * DISPLAY'WIDTH + H'SPACE'2 GR *---*STUMP'REC = FATHER $ *
           * * PAGE'WIDTH $                 *    *DONE = FALSE $       *
           * ********************************    ***********************
           *
           *
           * ***********    ************************
           +-*ORIF 1 $ *----*DISPLAY'ROOM = TRUE $ *
             ***********    ************************
```

```
****************
* 145 FROM 138 *
****************
 *
 *
 *
********************************
*TYPE = VERT $                 *
*FATHER = BACK'V($ACCESS3(STUMP*
*'REC)$) $                     *
*DISPLAY'ROOM = TRUE $         *
*'' SHOULD HAVE ENOUGH ROOM '' *
********************************
```

```
****************
* 146 FROM 140 *
****************
 *
 *
 *
*************************
*LAST*INDEX = INDEX $    *
*INDEX = NEXT($INDEX$) $ *
*************************
```

```
C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER                                  PAGE 147
DESIGN DIAGRAM OF RESOLVE'STUMP

****************
* 147 FROM 141 *
****************
 *
 *
 *
**************************
*LAST'INDEX = INDEX $    *
*INDEX = NEXT($INDEX$) $ *
**************************
```

```
DESIGN DIAGRAM OF RESOLVE'STUMP

****************
* 148 FROM 142 *
****************
 *
 *
 *
********************************
*TRAVERSE'TOP = TRAVERSE'TOP - *
* 1 $ TRAVERSE'TOP = TRAVERSE'T*
*STUMP'REC = TRAVERSE'STACK(STR*
*AVERSE'TOPS) $                *
********************************
```

DESIGN DIAGRAM OF STARS

```
********************************
*PROC STARS(COL, LNGTH, LINE) $ *
*                                *
*                               *
********************************
 *
 *
 *
 * '' THE PROCEDURE STARS OUTPUTS
 *    A STRING OF LNGTH STARS
 *   STARTING ON COLUMN COL OF THE
 *   LINE TH PUTOUT RECORD. ''
 *
 *
********************************
*ITEM COL INTEGER $            *
*'' THE COLUMN IN WHICH STARS  *
* START''                      *
*ITEM LINE INTEGER $           *
*''THE PUTOUT RECORD IN WHICH  *
* STARS APPEAR''               *
*ITEM LNGTH INTEGER $          *
*''NUMBER OF STARS OUTPUT''    *
*ITEM STAR H 132 P             *
* 132H(*************************
********************************
********************************
********************************
***        *****************) $ *
*BYTE($COL, LNGTH$)(OUT'LINE($A*
*CCESS'OUT(LINE)$)) = BYTE($0, *
* LNGTH$)(STAR) $              *
*RETURN $                      *
********************************
```

```
**************************
*PROC TRANSFER'WRITE3 $  *
**************************
 *
 *
 *
 * '' THE PROCEDURE
 *   TRANSFER'WRITE3 TRANSFERS THE
 *   CONTENTS OF THE MOST RECENT
 *   VERSION OF FILE 3 AND THE
 *   CURRENT BUFFER CONTENTS INTO
 *   THE SPARE VERSION OF FILE
 *   3.''
 * ''TMP'BUF IS A TEMP BUFFER FOR
 *   FILE 3''
 *
 *
*******************************
*TABLE TMP'BUF R F3'BLKSIZ S N*
* $                           *
*BEGIN                        *
*ITEM ENT1 INTEGER $          *
*ITEM ENT2 INTEGER`$          *
*ITEM ENT3 INTEGER $          *
*ITEM ENT4 INTEGER $          *
*ITEM ENT5 INTEGER $          *
*ITEM ENT6 INTEGER $          *
*ITEM ENT7 INTEGER $          *
*ITEM ENT8 INTEGER $          *
*IJEM ENT9 INTEGER $          *
*ITEM ENT10 INTEGER $         *
*ITEM ENT11 INTEGER $         *
*ITEM ENT12 INTEGER $         *
*ITEM ENT13 INTEGER $         *
*ITEM ENT14 INTEGER $         *
*END                          *
*ITEM TMPI INTEGER $          *
*'' TEMP STORE FOR            *
* RECS'IN'BLK3''              *
*******************************
 *
 *
***************
*IF DEBUG4 $  *    ********************************
*''7''        *---*MESSAGE = 11H(TRANS3-OUT ) $  *
*             *    *T'MESS = IFORMAT(F3'ƏLK) $    *
***************    *MESSAGE = CAT(MESSAGE, T'MESS)*
 *                 * $                            *
 *                 *OUT(MESSAGE, MESS'SW) $       *
 *                 ********************************
 *
****************************
*F3'EMPTY = MAX(F3'EMPTY,*
* F3'BLK + 1) $          *
****************************
```

```
C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER                                                     PAGE 151
DESIGN DIAGRAM OF TRANSFER'WRITE3

   **********   ***************
   *IFEITH  *-+-*NEW'FILE3 $  *-----+
   ********** * ***************     *  ** TRANSFER FROM  FILE3'2 TO
    *         *                     *  FILE3'1 **
    *         *                     *
    *         *                     *
    *         *                    ************************
    *         *                    *OPEN INPUT FILE3'2 $  *
    *         *                    *OPEN OUTPUT FILE3'1 $ *
    *         *                    ************************
    *         *                     *
    *         *                     *
    *         *                    ********************************   **********   *****************  ******
    *         *                    *FOR I = 0, 1, F3'EMPTY - 1 $  *---*IFEITH  *-+-*I EQ F3'BLK $  *--* 153*
    *         *                    ********************************   ********** * *****************  ******
    *         *                     *                                            *
    *         *                     *                                            *
    *         *                     *                                            * ************   ********************************
    *         *                     *                                            +-*ORIF 1 $  *---*POS(FILE3'2) = I $            *
    *         *                     *                                              ************   *INPUT FILE3'2 TMPI, TMP'BUF $ *
    *         *                     *                                                             *OUTPUT FILE3'1 TMPI, TMP'BUF $*
    *         *                     *                                                             *                              *
    *         *                     *                                                             ********************************
    *         *                     *
    *         *                    ************************
    *         *                    *SHUT INPUT FILE3'2 $  *
    *         *                    *SHUT OUTPUT FILE3'1 $ *
    *         *                    ************************
    *         *
    *         * ************
    *         +-*ORIF 1 $  *----+
    *           ************    *  ** TRANSFER FROM FILE3'1 TO
    *                           *  FILE3'2 **
    *                           *
    *                          ************************
    *                          *OPEN INPUT FILE3'1 $  *
    *                          *OPEN OUTPUT FILE3'2 $ *
    *                          ************************
    *                           *
    *                          ********************************   **********   *****************   ******************************
    *                          *FOR I = 0, 1, F3'EMPTY - 1 $  *---*IFEITH  *-+-*I EQ F3'BLK $  *---*OUTPUT FILE3'2 RECS'IN'BLK3,*
    *                          ********************************   ********** * *****************   * F3'BJF $                   *
    *                           *                                            *                     ******************************
    *                           *                                            *
    *                           *                                            * ************   ********************************
    *                           *                                            +-*ORIF 1 $  *---*POS(FILE3'1) = I $            *
    *                           *                                              ************   *INPUT FILE3'1 TMPI, TMP'BUF $ *
    *                           *                                                             *OUTPUT FILE3'2 TMPI, TMP'BUF $*
    *                           *                                                             * OUTPUT FILE3'2 TMPI, TMP'BUF *
    *                           *                                                             ********************************
    *                           *
    *                           *
    *                          ************************
    *                          *SHUT INPUT FILE3'1 $  *
    *                          *SHUT OUTPUT FILE3'2 $ *
```

DESIGN DIAGRAM OF TRANSFER"WRITE3

```
                        ************************

*****************************
*NEW"FILE3 = NOT NEW"FILE3 $ *
*WRITE3 = FALSE $            *
*RETURN $                    *
*****************************
```

```
****************
* 153 FROM 151 *
****************
 *
 *
 *
*******************************
*OUTPUT FILE3'1 RECS'IN'BLK3,*
* F3'BUF $                    *
*******************************
```

```
**************************
*PROC TRANSFER'WRITE4 $  *
**************************
 *
 *
 *
 * ''SAME AS TRANSFER'WRITE 3 ,
 *  ONLY IT OPERATES ON FILE 4
 *  TMP'BUF IS THE TEMP FILE 4
 *  BUFFER''
 *
 *
*********************************
*TABLE TMP'BUF R F4'BLKSIZ S N*
* $                             *
*BEGIN                          *
*ITEM ENT1 INTEGER $            *
*ITEM ENT2 INTEGER $            *
*ITEM ENT3 INTEGER $            *
*END                            *
*ITEM TMPI INTEGER $            *
*''TEMP STORE FOR               *
* RECS'IN'BLK4''                *
*********************************
 *
 *
***************
*IF DEBUG5 $  *      ************************************
*''7''         *---*MESSAGE = 11H(TRANS4-OUT ) $    *
*             *      *T'MESS = IFORMAT(F4'BLK) $        *
***************      *MESSAGE = CAT(MESSAGE, T'MESS)*
 *                   * $                                *
 *                   *OUT(MESSAGE, MESS'SW) $           *
 *                   ************************************
 *
****************************
*F4'EMPTY = MAX(F4'EMPTY,*
* F4'BLK + 1) $            *
****************************
 *
 *
**********     ***************
*IFEITH  *-+-*NEW'FILE4 $  *----+
**********  *  ***************     * '' TRANSFER FROM  FILE4'2 TO
 *          *                      *   FILE4'1 ''
 *          *                      *
 *          *                      *
 *          *                     **************************
 *          *                     *OPEN INPUT FILE4'2 $  *
 *          *                     *OPEN OUTPUT FILE4'1 $ *
 *          *                     **************************
 *          *                      *
 *          *                      *
 *          *                     ***********************************     **********     *****************     ******
 *          *                     *FOR I = 0, 1, F4'EMPTY - 1 $    *---*IFEITH   *-+-*I EQ F4'BLK $   *--* 156*
 *          *                     ***********************************     **********  *  *****************     ******
 *         *                    *                                           * ***********     *******************************
 *         *                    *                                           +--*ORIF 1 $  *---*POS(FILE4'2) = I $            *
 *         *                    *                                             ***********     *INPUT FILE4'2 TMPI, TMP'BUF $ *
 *         *                    *                                                             *OUTPUT FILE4'1 TMPI, TMP'BUF $*
 *         *                    *                                                             *                              *
 *         *                    *                                                             *******************************
 *         *                    *
 *         *                  ***********************
 *         *                  *SHUT INPUT FILE4'2 $  *
 *         *                  *SHUT OUTPUT FILE4'1 $ *
 *         *                  ***********************
 *         *
 *         *  ***********
 *         +--*ORIF 1 $  *----+
 *            ***********     *  ** TRANSFER FROM FILE4'1 TO
 *                            *   FILE4'2 **
 *                            *
 *                           ***********************
 *                           *OPEN INPUT FILE4'1 $  *
 *                           *OPEN OUTPUT FILE4'2 $ *
 *                           ***********************
 *                            *
 *                           ******************************   *********   ***************     ******************************
 *                           *FOR I = 0, 1, F4'EMPTY - 1 $ *---*IFEITH  *-+--*I EQ F4'BLK $ *---*OUTPUT FILE4'2 RECS'IN'BLK4,*
 *                           ******************************   *********  *  ***************    * F4'BUF $                   *
 *                            *                                          *                     ******************************
 *                            *                                          *  ***********      ******************************
 *                            *                                          +--*ORIF 1 $  *---*POS(FILE4'1) = I $           *
 *                            *                                             ***********      *INPUT FILE4'1 TMPI, TMP'BUF $ *
 *                            *                                                              *OUTPUT FILE4'2 TMPI, TMP'BUF $*
 *                            *                                                              *                              *
 *                            *                                                              ******************************
 *                            *
 *                           ***********************
 *                           *SHUT INPUT FILE4'1 $  *
 *                           *SHUT OUTPUT FILE4'2 $ *
 *                           ***********************
 *
*****************************
*NEW'FILE4 = NOT NEW'FILE4 $ *
*WRITE4 = FALSE $            *
*RETURN $                    *
*****************************
```

DESIGN DIAGRAM OF TRANSFER'WRITE4

```
****************
* 156 FROM 154 *
****************
*
*
*
*******************************
*OUTPUT FILE4'1 RECS'IN'SLK4,*
* F4'BUF $                    *
*******************************
```

```
****************************
*PROC TRANSFER'WRITE'OUT $  *
****************************
 *
 *
 *
 * '' SAME AS TRANSFER'WRITE3,
 *   EXCEPT TRANSFER'WRITE'OUT
 *   OPERATES ON PUTOUT. ''
 *
 *
***************
*IF DEBUG6 $  *    *********************************
*''7''         *---*MESSAGE = 10H(TRANS OUT ) $    *
*             *    *T'MESS = IFORMAT(OUT'BLK) $    *
***************    *MESSAGE = CAT(MESSAGE, T'MESS)*
 *                 * $                             *
 *                 *OUT(MESSAGE, MESS'SW) $        *
 *                 *********************************
 *
********************************
*MAX'FOUTPUT = MAX(MAX'FOUTPUT,*
* OUT'BLK) $                   *
********************************
 *
 *
*********   ************
*IFEITH  *-+-*NEW'OUT $  *----+
*********  * ************     * '' TRANSFER FROM PUTOUT'2 TO
 *         *                  *  PUTOUT'1 ''
 *         *                  *
 *         *                  *
 *         *                 **************************
 *         *                 *OPEN INPUT PUTOUT'2 $  *
 *         *                 *OPEN OUTPUT PUTOUT'1 $ *
 *         *                 **************************
 *         *                  *
 *         *                  *
 *         *                 ******************************     *********   ****************   ******
 *         *                 *FOR I = 0, 1, MAX'FOUTPUT $  *---*IFEITH  *-+-*I EQ OUT'BLK $  *--* 159*
 *         *                 ******************************     *********  * ****************   ******
 *         *                  *                                            *
 *         *                  *                                            *
 *         *                  *                                            * ***********     *********************
 *         *                  *                                            +-*ORIF 1 $  *---*POS(PUTOUT'2) = I **
 *         *                  *                                              ***********     * OUT'BUF'SIZE $    *
 *         *                  *                                                              *********************
 *         *                  *                                                               *
 *         *                  *                                                               *
 *         *                  *                                                               *
 *         *                  *                                                              ******
 *         *                  *                                                              * 160*
 *         *                  *                                                              ******
 *         *                 **************************
 *         *                 *SHUT INPUT PUTOUT'2 $  *
```

```
C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER                                                          PAGE 158
DESIGN DIAGRAM OF TRANSFER'WRITE'OUT
                             *SHUT OUTPUT PUTOUT'1 $ *
 .         .                 *************************
 .         .
 .         . ************
 .         +--*ORIF 1 $  *-----+
 .           ************      . ** TRANSFER FROM OUTPUT'1 TO
 .                             .  PUTOUT'2 **
 .                             .
 .                             .
 .                            *************************
 .                            *OPEN INPUT PUTOUT'1 $  *
 .                            *OPEN OUTPUT PUTOUT'2 $ *
 .                            *************************
 .                             .
 .                             .
 .                            ******************************    *********    ****************    ******
 .                            *FOR I = 0, 1, MAX'FOUTPUT $  *---*IFEITH  *-+-*I EQ OUT'BLK $  *--* 161*
 .                            ******************************    *********  . ****************    ******
 .                             .                                           .
 .                             .                                           .
 .                             .                                           . ***********     ***********************
 .                             .                                           +-*ORIF 1 $  *---*POS(PUTOUT'1) = I **
 .                             .                                             ***********     * OUT'BUF'SIZE $     *
 .                             .                                                             ***********************
 .                             .                                                              .
 .                             .                                                              .
 .                             .                                                              .
 .                             .                                                             ******
 .                             .                                                             * 162*
 .                             .                                                             ******
 .                            *************************
 .                            *SHUT INPUT PUTOUT'1 $  *
 .                            *SHUT OUTPUT PUTOUT'2 $ *
 .                            *************************
 .
**************************
*NEW'OUT = NOT NEW'OUT $ *
*EXTRA'BLOCK = FALSE $   *
**************************
 .
 .
**************
*IF DEBUG6 $ *    *********************************
*''7''        *---*MESSAGE = 12H(MAX'FOUTPUT ) $ *
*            *    *T'MESS = IFORMAT(MAX'FOUTPUT) *
**************    * $                            *
 .                *MESSAGE = CAT(MESSAGE, T'MESS)*
 .                * $                            *
 .                *OUT(MESSAGE, MESS'SW) $       *
 .                *********************************
 .
************
*RETURN $ *
***********
```

```
DESIGN DIAGRAM OF TRANSFER'WRITE'OUT

*****************
* 159 FROM 157 *
*****************
 *
 *
 *
**********************************
*FOR J = 0, 1, OUT'BUF'SIZE - 1 *    ********************************
* $                              *---*OUTPUT PUTOUT'1 OUT'LINE($J$)*
*                                *    * $                            *
**********************************    *OUT'LINE($J$) = 1H( ) $       *
                                      ********************************
```

```
C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER                                                       PAGE 160
DESIGN DIAGRAM OF TRANSFER*WRITE*OUT

****************
* 160 FROM 157 *
****************
 *
 *
 *
*********************************
*FOR J = 0, 1, OUT*BUF*SIZE - 1 *    **************************
* $                             *---*INPUT PUTOUT*2 TEMPC $  *
*                               *    *OUTPUT PUTOUT*1 TEMPC $ *
*********************************    **************************
```

```
DESIGN DIAGRAM OF TRANSFER'WRITE'OUT

****************
* 161 FROM 158 *
****************
 *
 *
 *
*********************************
*FOR J = 0, 1, OUT'BUF'SIZE - 1 *    *******************************
* $                              *---*OUTPUT PUTOUT'2 JUT'LINE($J$)*
*                               *    * $                           *
*********************************    *OUT'LINE($J$) = 1H( ) $      *
                                     *******************************
```

```
*****************
* 162 FROM 158 *
*****************
 *
 *
 *
*********************************
*FOR J = 0, 1, OUT*BUF*SIZE - 1 *     ****************************
* $                             *---*INPUT PUTOUT*1 TEMPC $  *
*                               *    *OUTPUT PUTOUT*2 TEMPC $ *
*********************************     ****************************
```

```
***********************
*PROC UPDATE'FILE3 $  *
***********************
 *
 *
 *
 * '' THE PROCEDURE UPDATE'FILE3
 *   UPDATES FILE 3 (BY ADDING A
 *   NEW RE ORD, OR CONTINUING AN
 *   OLD ONE) IN ACCORDANCE WITH
 *   THE INFORMATION IN THE NEWLY
 *   CREATED SET OF FILE 4 RECS.
 *   ''
 *
 *
********************************
*ITEM LA'TYPE INTEGER $        *
*'' STMT'TOKEN OF THE NEXT FILE*
* 1 RECORD''                   *
*ITEM NEW'REC INTEGER $        *
*''THE RECORD NUMBER OF A NEW  *
* FILE3 REC''                  *
*ITEM STMT'TYPE INTEGER $      *
*'' THE TYPE OF THE CURRENT    *
* FILE3 REC''                  *
********************************
 *
 *
 *
 * ''UPDATE FILE 3 (IE THE PRINT
 *   CHAIN WITH THE NEWLY  ''
 * '' CREATED FILE 4 RECORDS ''
 *
 *
********************************
*STMT'TYPE = BOX'MAP(STMT'TOKEN*
*) $                           *
********************************
 *
 *
**********   ****************************
*IFEITH  *-+-*STMT'TYPE EQ END'SCGPE $  *----+
********** * ****************************    * ''LOOK AHEAD TO SEE IF CUR'REC
 *         *                                 *  MUST BE CONTINUED TO ''
 *         *                                 * '' INCLUDE A TYPE 1 COMMENT.
 *         *                                 * ''
 *         *                                 *
 *         *                                 *
 *         *                                ****************
 *         *                                *IF DEBUG10 $  *    *********************************
 *         *                                *''5''          *---*MESSAGE = 12H(UP END SCOPE) $ *
 *         *                                *               *   *OUT(MESSAGE, MESS'SW) $        *
 *         *                                ****************    *********************************
 *         *                                 *
 *         *                                 *
 *         *                                ****************
```

```
.         .                                ***************
.         .                                 .
.         .                                 .
.         .                                ******************    *******************************
.         .                                *IF NOT EOFILE $  *---*LA'TYPE = BOX'MAP(STMT'TOKEN)*
.         .                                ******************    * $                           *
.         .                                 .                    *******************************
.         .                                 .                     .
.         .                                 .                     .
.         .                                 .                     .
.         .                                 .                    ******
.         .                                 .                    * 168*
.         .                                 .                    ******
.         .                                 .
.         .                                ******************
.         .                                *IF NULL'SCOPE $  *----+
.         .                                ******************     . ** THE LAST CONTROL PHRASE HAS
.         .                                 .                     .   NO SCOPE -- CREATE A **
.         .                                 .                     . ** DUMMY BOX RECORD **
.         .                                 .                     .
.         .                                 .                     .
.         .                                 .                     .
.         .                                 .                    ******
.         .                                 .                    * 169*
.         .                                 .                    ******
.         .                                 .
.         .                                 .  ** FINISH UP CUR'REC **
.         .                                 .
.         .                                 .
.         .                                ********************************
.         .                                *IF STMT'UNIT($ACCESS3(CUR'REC) *     **********************
.         .                                *$) GQ 0 $                       *---*CLOSE'REC(CUR'REC) $ *
.         .                                *                              *      **********************
.         .                                ********************************
.         .                                 .
.         .                                 .
.         .                                 .
.         .                                 .  ** RETURN TO CONTROL PHRASE
.         .                                 .    BEING TERMINATED **
.         .                                 .
.         .                                 .
.         .                                ********************************
.         .                                *DO WHILE [BACK'H($ACCESS3(CUR' *    ******
.         .                                *REC)$) EQ 0 AND CUR'REC NQ 0]   *--* 170*
.         .                                *                              *     ******
.         .                                ********************************
.         .                                 .
.         .                                 .
.         .                                **********    ******************
.         .                                *IFEITH  *-+-*CUR'REC EQ 0 $  *----+
.         .                                **********  . ******************    . ** A PROCEDURE OR CLOSE OR
.         .                                            .                       .   PROGRAM IS BEING TER1INATED
.         .                                            .                       .   **
.         .                                            .                       .
```

```
 .         .                                           .                      ******************************
 .         .                                           .                      *POP'PROC'STACK(= CUR'REC) $ *
 .         .                                           .                      *POP'LAYOUT'INFO $           *
 .         .                                           .                      ******************************
 .         .                                           .
 .         .                                           . ***********  ******
 .         .                                           +-*ORIF 1 $  *--* 171*
 .         .                                             ***********  ******
 .         .
 .         .
 .         . ********************************
 .         +-*ORIF STMT'TYPE EQ COMMENT'1 $  *----+
 .         . ********************************     *  '' INCLUDE THE COMMENT IN
 .         .                                      *   WHATEVER BOX WE'RE IN ''
 .         .                                      *
 .         .                                      *
 .         .                                     **************************
 .         .                                     *CONTINUE'BOX(CUR'REC) $ *
 .         .                                     **************************
 .         .
 .         . ********************************
 .         +-*ORIF STMT'TYPE EQ CONTROL'1 $  *----+
 .         . ********************************     *  '' A PROCEDURE OR CLOSE OR
 .         .                                      *   PROGRAM HEAD ''
 .         .                                      *
 .         .                                      *
 .         .                                     ***************
 .         .                                     *IF DEBUG10 $  *    *******************************
 .         .                                     *''5''         *---*MESSAGE = 11H(UP CONTROL1) $ *
 .         .                                     *              *    *OUT(MESSAGE, MESS'SW) $       *
 .         .                                     ***************     *******************************
 .         .                                      *
 .         .                                      *
 .         .                                     ********************************
 .         .                                     *IF PROC'STACK'TOP GR 0 AND     *    ************************
 .         .                                     * STMT'UNIT($ACCESS3(CUR'REC)$) *---*CLOSE'REC(CUR'REC) $ *
 .         .                                     * GQ 0 $                        *    ************************
 .         .                                     ********************************
 .         .                                      *
 .         .                                      *
 .         .                                     *******************************
 .         .                                     *PUSH'PROC'STACK(CUR'REC) $   *
 .         .                                     *PUSH'LAYOUT'INFO $           *
 .         .                                     *CUR'REC = F3'AVAIL $         *
 .         .                                     *F3'AVAIL = F3'AVAIL + 1 $    *
 .         .                                     *INSERT(PROC'ROOT, CUR'REC) $ *
 .         .                                     *******************************
 .         .                                      *
 .         .                                      *
 .         .                                      *  '' NULL OUT POINTERS ''
 .         .                                      *
 .         .                                      *
 .         .                                     ********************************
 .         .                                     *BACK'H($ACCESS3(CUR'REC)$) = 0*
 .         .                                     * $                            *
 .         .                                     *BACK'V($ACCESS3(CUR'REC)$) = 0*
```

```
C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DI* $ MMER                       *                                  PAGE 166
DESIGN DIAGRAM OF UPDATE'FILE3                   *H'PTR($ACCESS3(CUR'REC)$) = 0 *
                                                 * $                            *
                                                 *V'PTR($ACCESS3(CUR'REC)$) = 0 *
 .         .                                     * $                            *
 .         .                                     *INITIATE'RECORD(CUR'REC) $    *
 .         .                                     *NULL'SCOPE = TRUE $           *
 .         .                                     *WRITE3 = TRUE $               *
 .         .
 .         .
 .         .
 .         .                                     ********************************
 .         .
 .         . *********************************
 .         +-*ORIF(/ STMT'UNIT($ACCESS3(CUR' *
 .         . *REC)$) /) GQ CONTROL'1 $        *----+
 .         . *                               *     * ** NEW STMT UNIT  STARTS SCOPE
 .         . *********************************     *  OF A CONTROL PHRASE **
 .         .                                       *
 .         .                                       *
 .         .                                      ****************
 .         .                                      *IF DEBUG10 $  *    *******************************
 .         .                                      ***5**          *---*MESSAGE = 14H(UP SCOPE START)*
 .         .                                      *              *    * $                           *
 .         .                                      ****************    *OUT(MESSAGE, MESS'SW) $      *
 .         .                                       .                  *******************************
 .         .                                       .
 .         .                                      **********   *********************************
 .         .                                      *IFEITH  *-+-*(/ STMT'UNIT($ACCESS3(CUR'REC) *
 .         .                                      ********** * *$) /) EQ CONTROL'1 OR          *    ******
 .         .                                       .         * * H'PTR($ACCESS3(CUR'REC)$) NQ  *--* 172*
 .         .                                       .         * * 0 $                           *    ******
 .         .                                       .         . *                               *
 .         .                                       .         . *********************************
 .         .                                       .         .
 .         .                                       .         .
 .         .                                       .         . ***********   ****************************
 .         .                                       .         +-*ORIF 1 $ *---*CREATE'H'PTR'REC(CUR'REC =*
 .         .                                       .           ***********   * NEW'REC) $                *
 .         .                                       .                         ****************************
 .         .                                       .
 .         .                                      *********************************
 .         .                                      *IF STMT'UNIT($ACCESS3(CUR'REC) *   ***********************
 .         .                                      *$) GQ 0 $                      *---*CLOSE'REC(CUR'REC) $ *
 .         .                                      *                               *   ***********************
 .         .                                      *********************************
 .         .                                       .
 .         .                                       .
 .         .                                      *****************************
 .         .                                      *CUR'REC = NEW'REC $        *
 .         .                                      *INITIATE'RECORD(CUR'REC) $ *
 .         .                                      *****************************
 .         .                                       .
 .         .                                       .
 .         .                                      **********   ***************************   **********************
 .         .                                      *IFEITH  *-+-*STMT'TYPE GQ CONTROL'1 $ *---*NULL'SCOPE = TRUE $ *
 .         .                                      ********** . ***************************   **********************
 .         .                                                 .
 .         .                                                 .
 .         .                                                 .
```

```
DESIGN DIAGRAM OF UPDATE*FILE3

*         *
*         *                                                 * ********************************    **********************
*         *                                                 +-*ORIF STMT*TYPE NQ COMMENT*2 $  *---*NULL*SCOPE = FALSE $ *
*         *                                                   ********************************    **********************
*         *
*         * ********************************
*         +-*ORIF STMT*TYPE NQ              *
*         * * STMT*UNIT($ACCESS3(CUR*REC)$) *----+
*         * * $                             *    * ** INITIATE A NEW BLOCK-- HANG
*         * ********************************    *    IT FORM V*PTR **
*         *                                      *
*         *                                     ***************
*         *                                     *IF DEBUG18 $ *    ******************************
*         *                                     ***5*         *---*MESSAGE = 11H(UP NEW TYPE) $ *
*         *                                     *             *   *OUT(MESSAGE, MESS*SW) $      *
*         *                                     ***************    ******************************
*         *                                      *
*         *                                     ****************************
*         *                                     *CREATE* V*PTR*REC(CUR*REC =*
*         *                                     * NEW*REC) $               *
*         *                                     ****************************
*         *                                      *
*         *                                     ********************************
*         *                                     *IF STMT*UNIT($ACCESS3(CUR*REC) *   ***********************
*         *                                     *$) GQ 0 $                      *---*CLOSE*REC(CUR*REC) $ *
*         *                                     *                               *   ***********************
*         *                                     ********************************
*         *                                      *
*         *                                     *****************************
*         *                                     *CUR*REC = NEW*REC $        *
*         *                                     *INITIATE*RECORD(CUR*REC) $ *
*         *                                     *****************************
*         *                                      *
*         *                                     *********   ***************************     *********************
*         *                                     *IFEITH *-+-*STMT*TYPE GR CONTROL*1 $  *---*NULL*SCOPE = TRUE $ *
*         *                                     *********  * ***************************     *********************
*         *                                                *
*         *                                                * ********************************    **********************
*         *                                                +-*ORIF STMT*TYPE NQ COMMENT*2 $  *---*NULL*SCOPE = FALSE $ *
*         *                                                  ********************************    **********************
*         *
*         * ***********    **************************
*         +-*ORIF 1 $ *---*CONTINUE*BOX(CUR*REC) $ *
*           ***********    **************************
*
***********
*RETURN $ *
***********
```

```
C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER                                    PAGE 168
DESIGN DIAGRAM OF UPDATE'FILE3
****************
* 168 FROM 167 *
****************
    *
    *
    *
*********   **************************
*IFEITH  *-+--*LA'TYPE EQ COMMENT'1 $  *----+
*********  *  **************************    *
           *                                * ''CREATE A FILE 4 REC FOR THE
           *                                *   TRAILING COMMENT ''
           *                                *
           *                                *
           *                                ***************************
           *                                *CREATE'FILE4'RECS $      *
           *                                *CONTINUE'BOX(CUR'REC) $  *
           *                                ***************************
           *
           *  ***********    **************************
           +--*ORIF 1 $  *---*LAST'F1 = LAST'F1 - 1 $ *
              ***********    **************************
```

```
****************
* 169 FROM 167 *
****************
 *
 *
 *
*********    ********************************
*IFEITH  *-+-*STMT*UNIT($ACCESS3(CUR*REC)$) *
*********  * * GR CONTROL*1 $                *----+
 *         * *                              *     * ** H*PTR WILL POINT AT DUMMY
 *         * ********************************     *  **
 *         *                                      *
 *         *                                      *
 *         *                                     *****************************
 *         *                                     *CREATE*H*PTR*REC(CUR*REC =*
 *         *                                     * NEW*REC) $                *
 *         *                                     *****************************
 *         *                                      *
 *         *                                      *
 *         *                                     *********************************
 *         *                                     *IF STMT*UNIT($ACCESS3(CUR*REC) *    ***********************
 *         *                                     *$) GQ 0 $                       *---*CLOSE*REC(CUR*REC) $ *
 *         *                                     *                               *    ***********************
 *         *                                     *********************************
 *         *
 *         * ***********
 *         +-*ORIF 1 $  *----+
 *           ***********     * ** COMMENTS ARE IN THE NULL
 *                           *  SCOPE -- V *PTR WILL **
 *                           * ** POINT AT DUMMY RECORD **
 *                           *
 *                           *
 *                          *********************************
 *                          *IF STMT*UNIT($ACCESS3(CUR*REC) *    ***********************
 *                          *$) GQ 0 $                       *---*CLOSE*REC(CUR*REC) $ *
 *                          *                               *    ***********************
 *                          *********************************
 *                           *
 *                           *
 *                          ****************************
 *                          *CREATE*V*PTR*REC(CUR*REC =*
 *                          * NEW*REC) $               *
 *                          ****************************
 *
********************************
*LINES($ACCESS3(NEW*REC)$) = 1 *
* $                            *
*BLOCK*WIDTH($ACCESS3(NEW*REC)$*
*) = 1 $                       *
*STMT*UNIT($ACCESS3(NEW*REC)$) *
* = 0 $                        *
*CUR*REC = NEW*REC $           *
*NULL*SCOPE = FALSE $          *
*WRITE3 = TRUE $               *
********************************
```

```
****************
* 170 FROM 167 *
****************
 *
 *
 *
********************************
*IF BACK'V($ACCESS3(CUR'REC)$) *
* LS 0 $                        *----+
*                              *     : ** BACK THROUGH A STUMP **
********************************     *
 *                                   *
 *                                ********************
 *                                *POP'LAYOUT'INFO $ *
 *                                ********************
 *
*********************************
*CUR'REC =(/ BACK'V($ACCESS3(CU*
*R'REC)$) /) $                  *
*********************************
 *
 *
****************
*IF DEBUG10 $  *     *******************************
***7**          *---*T'MESS = IFORMAT(CUR'REC) $   *
*              *     *MESSAGE = 10H(BACK THRU ) $   *
****************     *MESSAGE = CAT(MESSAGE, T'MESS)*
                     * $                            *
                     *OUT(MESSAGE, MESS'SW) $       *
                     *******************************
```

```
C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER                                                        PAGE 171
DESIGN DIAGRAM OF UPDATE'FILE3

****************
* 171 FROM 167 *
****************
 :
 :
 :
********************************
*IF BACK'H($ACCESS3(CUR'REC)$) *
* LS 0 $                        *----+
*                              *     : ** BACK THROUGH STUMP **
********************************     :
 :                                   :
 :                                 ********************
 :                                 *POP'LAYOUT'INFO $ *
 :                                 ********************
 :
********************************
*CUR'REC =(/ BACK'H($ACCESS3(CU*
*R'REC)$) /) $                 *
********************************
```

```
DESIGN DIAGRAM OF UPDATE'FILE3

****************
* 172 FROM 167 *
****************
  *
  *
  *
****************************
*CREATE'V'PTR'REC(CUR'REC =*
* NEW'REC) $               *
****************************
```

DESIGN DIAGRAM OF OUT

```
*********************
*PROC OUT(AA, CC) $  *
*********************
 .
 .
 .
 .  **PUTS CONTENTS OF CHARACTER
 .   VARIABLE AA OUT TO THE ERROR
 .   FILE. IF CC IS 1, THIS FILE
 .   IS THE TERMINAL.  IF NOT, IT
 .   IS FILE12**
 .
 .
*******************
*ITEM CC I 36 S $ *
*ITEM AA H 150 $  *
*ITEM BB H 150 $  *
*ITEM DD H 80 $   *
*SF4 = AA $       *
*******************
 .
 .
********************************
*IF BYTE($0, 1$)(AA) NQ 1H([) $ *     *********************
*                                *--->*SF4 = CNVERT(AA) $ *
*                               *     *********************
********************************
 .
 .
*********************
*LN4 = LN4 - CONST $ *
*********************
 .
 .
**********   ************   *********************************
*IFEITH  *-+-*CC EQ 1 $  *---*BB = 1H( ) $                   *
**********  . ************   *BYTE($0, LN4$)(BB) = BYTE($6,*
            .                * LN4$)(SF4) $                  *
            .                *********************************
            .                 .
            .                 .
            .                **********   ***************   **********************************
            .                *IFEITH  *-+-*LN4 GR 72 $  *---*TRMOUT(BB, 72) $                *
            .                **********  . ***************   *LN4 = LN4 - 72 $                *
            .                            .                   *BYTE($0, LN4$)(BB) = BYTE($72,*
            .                            .                   * LN4$)(BB) $                    *
            .                            .                   *TRMOUT(BB, LN4) $               *
            .                            .                   **********************************
            .                            .
            .                            . *******************   **********************************
            .                            +-*ORIF LN4 EQ 0 $  *---*TRMOUT(13H(.....NIL.....), 13)*
            .                            . *******************   * $                              *
            .                            .                       **********************************
            .                            .
            .                            .
```

```
DESIGN DIAGRAM OF OUT

           .                            . ************   *********************
           .                            +-*ORIF 1 $  *---*TRMOUT(BB, LN4) $ *
           .                              ************   *********************
           .
           . ************    ****************
           +-*ORIF 1 $  *---*DD = 1H( ) $ *
             ************    ****************
                            .
                            .
                            **********  ***************     ******************************
                            *IFEITH  *-+-*LN4 GR 80 $  *---*DD = BYTE($6, 80$)(SF4) $ *
                            ********** . ***************    ******************************
                            .          .                    .
                            .          .                    .
                            .          .                    *********************   *********************************
                            .          .                    *IF INITST EQ 1 $  *---*INITST = 0 $                   *
                            .          .                    *********************   ***MUST OPEN FILE FOR OUTPUT ***
                            .          .                    .                       *                               *
                            .          .                    .                       *OPEN OUTPUT ZZZZZZ $           *
                            .          .                    .                       *********************************
                            .          .                    .
                            .          .                    *******************************
                            .          .                    *OUTPUT ZZZZZZ DD $           *
                            .          .                    *LN4 = LN4 - 80 $             *
                            .          .                    *DD = 1H( ) $                 *
                            .          .                    ***MUST CLEAR OUT STRING AGAIN *
                            .          .                    * FOR NEXT LINE**             *
                            .          .                    *BYTE($0, LN4$)(DD) = BYTE($86,*
                            .          .                    * LN4$)(SF4) $                *
                            .          .                    *******************************
                            .          .
                            .          . ******************    ***********************
                            .          +-*ORIF LN4 EQ 0 $  *---*BYTE($0, 13$)(DD) =  *
                            .          . ******************    * 13H(.....NIL.....) $*
                            .          .                       ***********************
                            .          .
                            .          . ************    *******************************
                            .          +-*ORIF 1 $  *---*BYTE($0, LN4$)(DD) = BYTE($6,*
                            .            ************    * LN4$)(SF4) $                *
                            .                            *******************************
                            .
                            *********************   *********************************
                            *IF INITST EQ 1 $  *---*INITST = 0 $                   *
                            *********************   ***MUST OPEN FILE FOR OUTPUT ***
                            .                       *                               *
                            .                       *OPEN OUTPUT ZZZZZZ $           *
                            .                       *********************************
                            .
                            ******************   ***********************
                            *IF LN4 GR 0 $  *---*OUTPUT ZZZZZZ DD $ *
                            ******************   ***********************
```

```
********************************
*PROC SUBSTR(AA, FIRST, NUM) $ *
********************************
 *
 *
 *
 *  **RETURNS THE SUBSTRING
 *   DEFINED BY CHARACTER STRING
 *   AA, INDEX *FIRST*, AND LENGTH
 *   NUM**
 *
*********************
*ITEM FIRST I 36 S $ *
*ITEM NUM I 36 S $   *
*ITEM AA H 150 $     *
*ITEM SUBSTR H 150 $ *
*SF1 = AA $          *
*SF2 = 1H( ) $       *
*********************
 *
 *
******************************
*IF BYTE($0, 1$)(SF1) NQ 1H(I) *    ********************
* $                            *---*SF1 = CNVERT(AA) $ *
*                              *    ********************
******************************
 *
 *
****************************     ****************
*IF NUM LQ 0 OR LN1 EQ 0 $  *---*LN1 = CONST $  *
****************************     *SUBSTR = SF1 $ *
 *                               *RETURN $       *
 *                               ****************
 *
*****************************
*IF FIRST + NUM - 1 GR (LN1 - *   ******************************
* CONST) $                    *---*TC = 20H(*** SUBSTR ERROR ***)*
*                             *   * $                            *
*****************************     *OUT(TC, RPTERR) $             *
 *                                *TC = 8H(STRING: ) $           *
 *                                *OUT(TC, RPTERR) $             *
 *                                *OUT(SF1, RPTERR) $            *
 *                                *TC = 18H(LENGTH OF STRING: ) $*
 *                                *                              *
 *                                *LN1 = LN1 - CONST $           *
 *                                *ENCODE(6H((  I6)), LN1 = TC6) *
 *                                * $                            *
 *                                *TC1 = TC6 $                   *
 *                                *OUT(CAT(TC, TC1), RPTERR) $   *
 *                                *TC = 7H(INDEX: ) $            *
 *                                *ENCODE(6H((  I6)), FIRST =    *
 *                                * TC6) $                       *
 *                                *TC1 = TC6 $                   *
 *                                *OUT(CAT(TC, TC1), RPTERR) $   *
 *                                *TC = 8H(LENGTH: ) $           *
 .                                 *ENCODE(6H((  I6)), NUM = TC6) *
 .                                 * $                            *
 .                                 *TC1 = TC6 $                   *
 .                                 *OUT(CAT(TC, TC1), RPTERR) $   *
 .                                 *LN1 = CONST $                 *
 .                                 *SUBSTR = SF1 $                *
 .                                 *RETURN $                      *
 .                                 ********************************
*******************************
*BYTE($6, NUM$)(SF2) = BYTE($5*
* + FIRST, NUM$)(SF1) $       *
*LN2 = NUM + CONST $          *
*SUBSTR = SF2 $               *
*******************************
```

```
DESIGN DIAGRAM OF CAT

*********************
*PROC CAT(AA, BB) $  *
*********************
 *
 *
 *
 * **RETURNS THE RESULT OF
 *   CONCATENATING STRING BB TO
 *   THE END OF STRING AA**
 *
 *
*******************
*ITEM AA H 150 $  *
*ITEM BB H 150 $  *
*ITEM CAT H 150 $ *
*SF1 = AA $       *
*SF2 = BB $       *
*******************
 *
 *
********************************
*IF BYTE($0, 1$)(SF1) NQ 1H(() *    *********************
* $                             *---*SF1 = CNVERT(AA) $ *
*                              *    *********************
********************************
 *
 *
********************************
*IF BYTE($0, 1$)(SF2) NQ 1H(() *    *********************
* $                             *---*SF2 = CNVERT(BB) $ *
*                              *    *********************
********************************
 *
 *
**************************    **************
*IF (LN2 - CONST) EQ 0 $  *---*CAT = SF1 $ *
**************************    *RETURN $    *
 *                            **************
 *
*****************************
*IF (LN1 + LN2 - CONST2) GR *     ********************************
* MAXCOL $                   *---*LN2 = MAXCOL - LN1 + CONST2 $ *
*                           *     *TC = 20H(*** CONCAT ERROR ***)*
*****************************     * $                            *
 *                                *OUT(TC, RPTERR) $             *
 *                                *TC = 18H(TRUNCATED STRING! ) $*
 *                                *                              *
 *                                *OUT(TC, RPTERR) $             *
 *                                *OUT(SF2, RPTERR) $            *
 *                                ********************************
 *
*********************************
*BYTE($6 + LN1 - CONST, LN2 -   *
* CONST$)(SF1) = BYTE($6, LN2 -*
* CONST$)(SF2) $                *
*LN1 = LN1 + LN2 - CONST $      *
```

DESIGN DIAGRAM OF CAT

```
*CAT = SF1 $                       *
************************************
```

```
*********************
*PROC CNVERT(AA) $  *
*********************
 .
 .
***********************
*ITEM AA H 150 $      *
*ITEM CNVERT H 150 $  *
*ITEM DONE I 36 S $   *
*ITEM II I 36 S $     *
*ITEM JJ I 36 S $     *
***********************
 .
 .
 .
 .  ''CNVERT CONVERTS NORMAL
 .   JOVIAL CHARACTER STRINGS OF
 .   LENGTH 150 TP A DESCRIPTOR
 .   FORM SUITABLE FOR EXTENDED
 .   STRING OPERATIONS. CNVERT
 .   WILL SIMPLY RETURN ITS
 .   ARGUMENT IF THE STRING IS
 .   ALREADY CONVERTED.  STRINGS
 .   CONTAINING MORE THA MAXCOL
 .   CHARACTERS WILL BE TRUNCATED
 .   ''
 .
 .
**************
*SF3 = AA $ *
**************
 .
 .
*********************************
*IF BYTE($0, 1$)(SF3) EQ 1H([) *    ******************
* $                             *---*CNVERT = SF3 $ *
*                               *    *RETURN $        *
*********************************    ******************
 .
 .
*********************************
*SA3($0$) = 6H(      ) $        *
*''PREVENTS OVERFLOW OF BYTE    *
* LATER ''                      *
*II = 1 $                       *
*DONE = 0 $                     *
*SF6 = 1H( ) $                  *
*''CLEARS OUT CHARACTER STRING*
* ''                            *
*********************************
 .
 .
 .
 .
 .
```

```
C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER                                                          PAGE 188
DESIGN DIAGRAM OF CNVERT

********************************
*DO WHILE (DONE EQ 0 AND II LQ *   *******************************   ***********
* 24)                          *---*IF SA3($II$) NQ 6H(      ) $ *---*JJ = 0 $ *
*                              *   *******************************   ***********
********************************    *                                 *
 *                                  *                                 *
 *                                  *                                *****************************
 *                                  *                                *DO WHILE (BYTE($JJ,        *   ****************
 *                                  *                                * 1$)(SA3($II$)) EQ 1H( ))  *---*JJ = JJ + 1 $ *
 *                                  *                                *                           *   ****************
 *                                  *                                *****************************
 *                                  *                                 *
 *                                  *                                 *
 *                                  *                                 *
 *                                  *                                 * ''FIRST NONBLANK CHARACTER HAS
 *                                  *                                 *  BEEN LOCATED EXACTLY ''
 *                                  *                                 *
 *                                  *                                 *
 *                                  *                                ********************************
 *                                  *                                *BYTE($6, 6 - JJ$)(SF6) =      *
 *                                  *                                * BYTE($JJ, 6 - JJ$)(SA3($II$))*
 *                                  *                                * $                            *
 *                                  *                                ********************************
 *                                  *                                 *
 *                                  *                                 *
 *                                  *                                 *
 *                                  *                                 * ''PICK UP THESE FIRST FEW
 *                                  *                                 *  WORD-SKEWED CHARACTERS''
 *                                  *                                 *
 *                                  *                                 *
 *                                  *                                **********************************
 *                                  *                                *IF II NQ 24 $                   *   ******
 *                                  *                                *''AVOID GIVING BYTE FUNCTION A  *---* 182*
 *                                  *                                * ZERO ARGUMENT'' ''AVOID GIVIN  *   ******
 *                                  *                                **********************************
 *                                  *                                 *
 *                                  *                                 *
 *                                  *                                 *
 *                                  *                                 * ''PICK UP REST OF STRING ''
 *                                  *                                 *
 *                                  *                                 *
 *                                  *                                ********************************
 *                                  *                                *LN6 = CONST + 150 - (II + 1) **
 *                                  *                                * 6 + 6 - JJ $                 *
 *                                  *                                ********************************
 *                                  *                                 *
 *                                  *                                 *
 *                                  *                                ****************************    ******
 *                                  *                                *IF LN6 GR MAXCOL + CONST $ *---* 183*
 *                                  *                                ****************************    ******
 *                                  *                                 *
 *                                  *                                 *
 *                                  *                                ****************************
 *                                  *                                *DONE = 1 $                *
 *                                  *                                *''SIGNAL END OF CONVERT '' *
 *                                  *                                ****************************
```

```
C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER                                                   PAGE 181
DESIGN DIAGRAM OF CNVERT

 *
 *                                    ******************
 *                                    *II = II + 1 $    *
 *                                    ***LOOP ON II **  *
 *                                    ******************
******************************    ***********************************
*IF II EQ 25 AND DONE EQ 0 $  *---*SF6 = SPACES(MAXCOL) $          *
******************************    ***LINE WAS ENTIRELY BLANK **    *
 *                                ***********************************
 *
*****************
*CNVERT = SF6 $ *
*****************
```

```
DESIGN DIAGRAM OF CNVERT

****************
* 182 FROM 180 *
****************
 *
 *
 *
********************************
*BYTE($12 - JJ, 144 - 6 *      *
* II$)(SF6) = BYTE($II * 6 + 6,*
* 144 - 6 * II$)(SF3) $        *
********************************
```

```
DESIGN DIAGRAM OF CNVERT

****************
* 183 FROM 180 *
****************
 *
 *
 *
***************************
*LN6 = MAXCOL + CONST $   *
*""TRUNCATE THE STRING "" *
***************************
```

DESIGN DIAGRAM OF SPACES

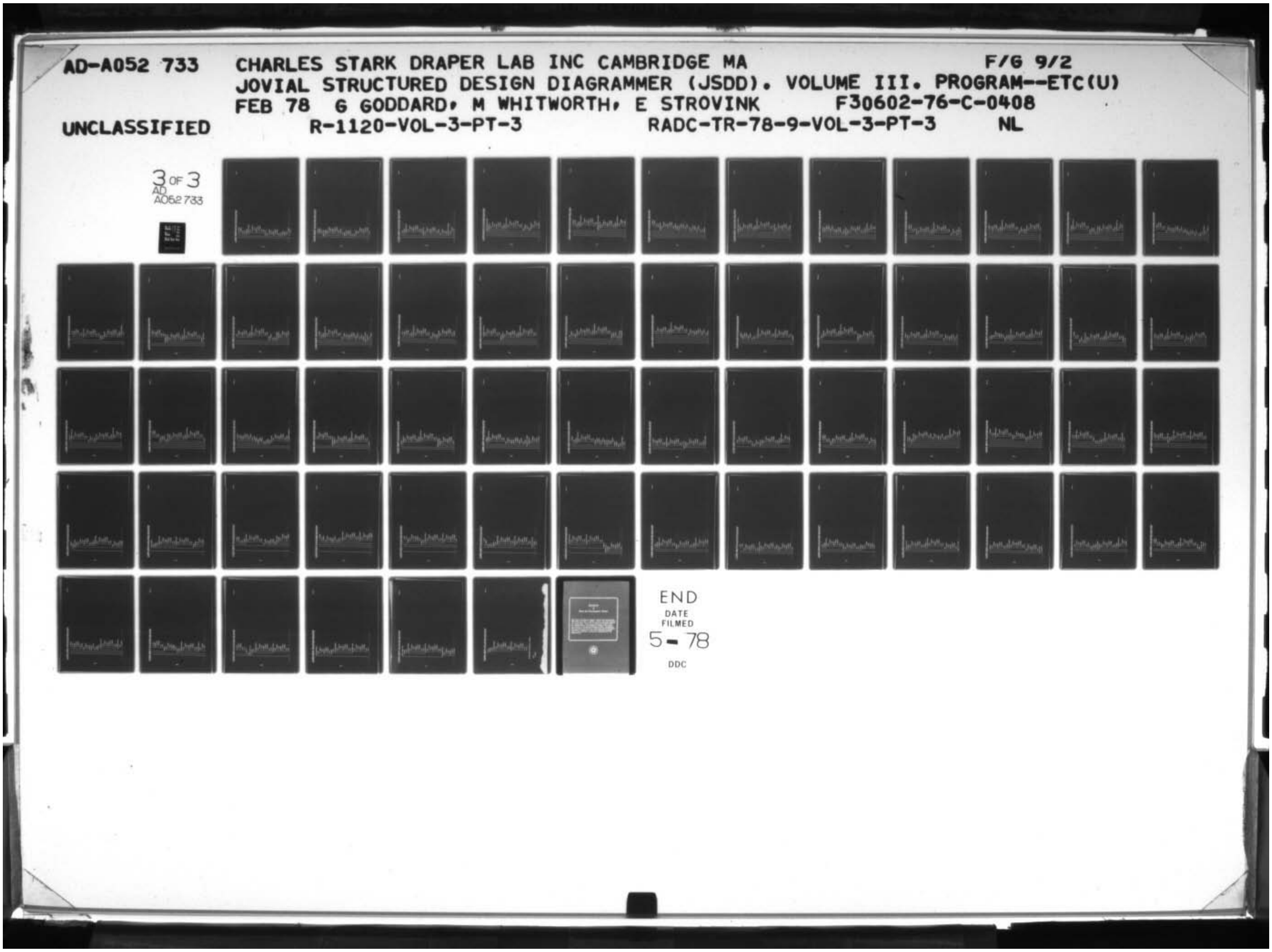
```
**********************
*PROC SPACES(NUM) $  *
**********************
 *
 *
 *
 *  **RETURNS AN EXTENDED
 *   CHARACTER STRING CONTAINING
 *   THE NUMBER OF SPACES DEFINED
 *   BY NUM**
 *
 *
**********************
*ITEM NUM I 36 S $   *
*ITEM SPACES H 150 $ *
**********************
 *
 *
**********************     *********************************
*IF NUM GR MAXCOL $  *---*NUM = MAXCOL $                   *
**********************    *TC = 20H(*** SPACES ERROR ***)*
 *                        * $                              *
 *                        *OUT(TC, RPTERR) $               *
 *                        *********************************
 *
**********************
*SF1 = 1H( ) $       *
*LN1 = CONST + NUM $ *
*SPACES = SF1 $      *
**********************
```

DESIGN DIAGRAM OF NULL

```
******************
*PROC NULL(AA) $  *
******************
 *
 *
 *
 *  **RETURNS TRUE IF STRING AA IS
 *   A NULL STRING**
 *
 *
******************
*ITEM AA H 150 $ *
*ITEM NULL B $   *
*SF1 = AA $      *
******************
 *
 *
*******************************
*IF BYTE($0, 1$)(SF1) NQ 1H(() *     **********************
* $                            *---*SF1 = CNVERT(SF1) $ *
*                              *     **********************
*******************************
 *
 *
*********    *****************     *************
*IFEITH  *-+-*LN1 EQ CONST $  *---*NULL = 1 $ *
*********  * *****************     *************
           *
           *
           * ************    *************
           +-*ORIF 1 $  *---*NULL = 0 $ *
             ************    *************
```

```
DESIGN DIAGRAM OF LENGTH

*********************
*PROC LENGTH(AA) $  *
*********************
 *
 *
 *
 *  **RETURNS THE LENGTH OF STRING
 *    AA**
 *
 *
***********************
*ITEM LENGTH I 36 S $ *
*ITEM AA H 150 $      *
*SF1 = AA $           *
***********************
 *
 *
********************************
*IF BYTE($0, 1$)(SF1) NQ 1H(() *     **********************
* $                             *---*SF1 = CNVERT(SF1) $ *
*                               *    **********************
********************************
 *
 *
*************************
*LENGTH = LN1 - CONST $ *
*************************
```

```
C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER                                    PAGE 1
INVOCATION DIAGRAM OF THE DESIGN DIAGRAM GENERATOR (DDG)

    ULTIMATELY SELF-RECURSIVE

    CNVERT
    OUT
    SPACES

     --***MAIN
       .
       *--BYTE*EN
       .  .
       .  *--RENQUO+
       .
       *--CAT
       .  .
       .  *--CNVERT
       .  .  .
       .  .  *--SPACES
       .  .     .
       .  .     *--OUT
       .  .        .
       .  .        *--CNVERT*
       .  .        .
       .  .        *--TRNOUT+
       .  .
       .  *--OUT*
       .
       *--CREATE*FILE4*RECS
       .  .
       .  *--ACCESS2
       .  .  .
       .  .  *--CAT
       .  .  .  .
       .  .  .  *--CNVERT*
       .  .  .  .
       .  .  .  *--OUT*
       .  .  .
       .  .  *--IFORMAT
       .  .  .  .
       .  .  .  *--CNVERT*
       .  .  .  .
       .  .  .  *--ENCODE+
       .  .  .
       .  .  *--OUT*
       .  .  .
       .  .  *--RENQUO+
       .  .
       .  *--ACCESS4
       .  .  .
       .  .  *--CAT
       .  .  .  .
       .  .  .  *--CNVERT*
       .  .  .  .
       .  .  .  *--OUT*
       .  .  .
       .  .  *--IFORMAT
       .  .  .  .
```



AD-A052 733 CHARLES STARK DRAPER LAB INC CAMBRIDGE MA F/G 9/2
JOVIAL STRUCTURED DESIGN DIAGRAMMER (JSDD). VOLUME III. PROGRAM--ETC(U)
FEB 78 G GODDARD, M WHITWORTH, E STROVINK F30602-76-C-0408
UNCLASSIFIED R-1120-VOL-3-PT-3 RADC-TR-78-9-VOL-3-PT-3 NL

3 OF 3
AD
A052733

END
DATE
FILMED
5-78
DDC

```
.  .  .  .--CNVERT*
.  .  .  .
.  .  .  .--ENCODE+
.  .  .
.  .  .--MAX
.  .  .
.  .  .--OUT*
.  .  .
.  .  .--REMQUO+
.  .  .
.  .  .--TRANSFER'WRITE+
.  .     .
.  .     .--CAT
.  .     .  .
.  .     .  .--CNVERT*
.  .     .  .
.  .     .  .--OUT*
.  .     .
.  .     .--IFORMAT
.  .     .  .
.  .     .  .--CNVERT*
.  .     .  .
.  .     .  .--ENCODE+
.  .     .
.  .     .--MAX
.  .     .
.  .     .--OUT*
.  .
.  .--BOX'MAP
.  .
.  .--CAT
.  .  .
.  .  .--CNVERT*
.  .  .
.  .  .--OUT*
.  .
.  .--IFORMAT
.  .  .
.  .  .--CNVERT*
.  .  .
.  .  .--ENCODE+
.  .
.  .--MAX
.  .
.  .--OUT*
.  .
.  .--SPACES*
.  .
.  .--SUBSTR
.     .
.     .--CAT
.     .  .
.     .  .--CNVERT*
.     .  .
.     .  .--OUT*
.     .
```

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER
INVOCATION DIAGRAM OF THE DESIGN DIAGRAM GENERATOR (DDG)

```
.     .--CNVERT*
.     .
.     .--ENCODE+
.     .
.     .--OUT*
.
.--GET'F1'REC
.  .
.  .--ACCESS1
.  .  .
.  .  .--CAT
.  .  .  .
.  .  .  .--CNVERT*
.  .  .  .
.  .  .  .--OUT*
.  .  .
.  .  .--IFORMAT
.  .  .  .
.  .  .  .--CNVERT*
.  .  .  .
.  .  .  .--ENCODE+
.  .  .
.  .  .--OUT*
.  .  .
.  .  .--REMQUO+
.  .
.  .--OUT*
.
.--IFORMAT
.  .
.  .--CNVERT*
.  .
.  .--ENCODE+
.
.--INITIALIZE
.
.--OUT*
.
.--PART2
.  .
.  .--ACCESS3
.  .  .
.  .  .--CAT
.  .  .  .
.  .  .  .--CNVERT*
.  .  .  .
.  .  .  .--OUT*
.  .  .
.  .  .--IFORMAT
.  .  .  .
.  .  .  .--CNVERT*
.  .  .  .
.  .  .  .--ENCODE+
.  .  .
.  .  .--MAX
.  .  .
.  .  .--OUT*
.  .  .
.  .  .--REMQUO+
.  .  .
.  .  .--TRANSFER'WRITE3
.  .  .  .
.  .  .  .--CAT
.  .  .  .  .
.  .  .  .  .--CNVERT*
.  .  .  .  .
.  .  .  .  .--OUT*
.  .  .  .
.  .  .  .--IFORMAT
.  .  .  .  .
.  .  .  .  .--CNVERT*
.  .  .  .  .
.  .  .  .  .--ENCODE+
.  .  .  .
.  .  .  .--MAX
.  .  .  .
.  .  .  .--OUT*
.  .
.  .--ADVANCE'PAGE
.  .  .
.  .  .--CAT
.  .  .  .
.  .  .  .--CNVERT*
.  .  .  .
.  .  .  .--OUT*
.  .  .
.  .  .--IFORMAT
.  .  .  .
.  .  .  .--CNVERT*
.  .  .  .
.  .  .  .--ENCODE+
.  .  .
.  .  .--OUT*
.  .  .
.  .  .--REMQUO+
.  .
.  .--CAT
.  .  .
.  .  .--CNVERT*
.  .  .
.  .  .--OUT*
.  .
.  .--CNVERT*
.  .
.  .--COMPUTE'PAGE'NUMBERS
.  .  .
.  .  .--CAT
.  .  .  .
.  .  .  .--CNVERT*
.  .  .  .
.  .  .  .--OUT*
.  .  .
```

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER
INVOCATION DIAGRAM OF THE DESIGN DIAGRAM GENERATOR (DDG)

```
.  .  '--GENERATE'CONTENTS'ENTRY
.  .  .  .
.  .  .  '--ACCESS'OUT
.  .  .  .  .
.  .  .  .  '--CAT
.  .  .  .  .  .
.  .  .  .  .  '--CNVERT*
.  .  .  .  .  .
.  .  .  .  .  '--OUT*
.  .  .  .  .
.  .  .  .  '--IFORMAT
.  .  .  .  .  .
.  .  .  .  .  '--CNVERT*
.  .  .  .  .  .
.  .  .  .  .  '--ENCODE+
.  .  .  .  .
.  .  .  .  '--OUT*
.  .  .  .  .
.  .  .  .  '--REMQUO+
.  .  .  .  .
.  .  .  .  '--TRANSFER'WRITE'OUT
.  .  .  .     .
.  .  .  .     '--CAT
.  .  .  .     .  .
.  .  .  .     .  '--CNVERT*
.  .  .  .     .  .
.  .  .  .     .  '--OUT*
.  .  .  .     .
.  .  .  .     '--IFORMAT
.  .  .  .     .  .
.  .  .  .     .  '--CNVERT*
.  .  .  .     .  .
.  .  .  .     .  '--ENCODE+
.  .  .  .     .
.  .  .  .     '--MAX
.  .  .  .     .
.  .  .  .     '--OUT*
.  .  .  .
.  .  .  '--CAT
.  .  .  .  .
.  .  .  .  '--CNVERT*
.  .  .  .  .
.  .  .  .  '--OUT*
.  .  .  .
.  .  .  '--DOTS
.  .  .  .  .
.  .  .  .  '--ACCESS'OUT
.  .  .  .     .
.  .  .  .     '--CAT
.  .  .  .     .  .
.  .  .  .     .  '--CNVERT*
.  .  .  .     .  .
.  .  .  .     .  '--OUT*
.  .  .  .     .
.  .  .  .     '--IFORMAT
```

INVOCATION DIAGRAM OF THE DESIGN DIAGRAM GENERATOR (DDG)

```
.  .  .  .     .  *--CNVERT*
.  .  .  .     .  .
.  .  .  .     .  *--ENCODE+
.  .  .  .     .
.  .  .  .     *--OUT*
.  .  .  .     .
.  .  .  .     *--REMQUO+
.  .  .  .     .
.  .  .  .     *--TRANSFER*WRITE*OUT
.  .  .  .        .
.  .  .  .        *--CAT
.  .  .  .        .  .
.  .  .  .        .  *--CNVERT*
.  .  .  .        .  .
.  .  .  .        .  *--OJT*
.  .  .  .        .       .
.  .  .  .        *--IFORMAT
.  .  .  .        .  .
.  .  .  .        .  *--CNVERT*
.  .  .  .        .  .
.  .  .  .        .  *--ENCODE+
.  .  .  .        .
.  .  .  .        *--MAX
.  .  .  .        .
.  .  .  .        *--OUT*
.  .  .  .
.  .  .  *--GENERATE*CONTENTS*HEADER
.  .  .  .  .
.  .  .  .  *--ACCESS*OUT
.  .  .  .  .  .
.  .  .  .  .  *--CAT
.  .  .  .  .  .  .
.  .  .  .  .  .  *--CNVERT*
.  .  .  .  .  .  .
.  .  .  .  .  .  *--OUT*
.  .  .  .  .  .
.  .  .  .  .  *--IFORMAT
.  .  .  .  .  .  .
.  .  .  .  .  .  *--CNVERT*
.  .  .  .  .  .  .
.  .  .  .  .  .  *--ENCODE+
.  .  .  .  .  .
.  .  .  .  .  *--OUT*
.  .  .  .  .  .
.  .  .  .  .  *--REMQUO+
.  .  .  .  .  .
.  .  .  .  .  *--TRANSFER*WRITE*OUT
.  .  .  .  .     .
.  .  .  .  .     *--CAT
.  .  .  .  .     .  .
.  .  .  .  .     .  *--CNVERT*
.  .  .  .  .     .  .
.  .  .  .  .     .  *--OUT*
.  .  .  .  .     .
.  .  .  .  .     *--IFORMAT
.  .  .  .  .     .  .
.  .  .  .  .     .  '--CNVERT*
.  .  .  .  .     .  .
.  .  .  .  .     .  '--ENCODE+
.  .  .  .  .     .
.  .  .  .  .     '--MAX
.  .  .  .  .     .
.  .  .  .  .     '--OUT*
.  .  .  .  .
.  .  .  .  '--ADVANCE'PAGE
.  .  .  .  .  .
.  .  .  .  .  '--CAT
.  .  .  .  .  .  .
.  .  .  .  .  .  '--CNVERT*
.  .  .  .  .  .  .
.  .  .  .  .  .  '--OUT*
.  .  .  .  .  .
.  .  .  .  .  '--IFORMAT
.  .  .  .  .  .  .
.  .  .  .  .  .  '--CNVERT*
.  .  .  .  .  .  .
.  .  .  .  .  .  '--ENCODE+
.  .  .  .  .  .
.  .  .  .  .  '--OUT*
.  .  .  .  .  .
.  .  .  .  .  '--REMQUO+
.  .  .  .  .
.  .  .  .  '--CNVERT*
.  .  .  .  .
.  .  .  .  '--IFORMAT
.  .  .  .  .  .
.  .  .  .  .  '--CNVERT*
.  .  .  .  .  .
.  .  .  .  .  '--ENCODE+
.  .  .  .  .
.  .  .  .  '--LENGTH
.  .  .  .  .  .
.  .  .  .  .  '--CNVERT*
.  .  .  .  .
.  .  .  .  '--OUT*
.  .  .  .
.  .  .  '--IFORMAT
.  .  .  .  .
.  .  .  .  '--CNVERT*
.  .  .  .  .
.  .  .  .  '--ENCODE+
.  .  .  .
.  .  .  '--LENGTH
.  .  .  .  .
.  .  .  .  '--CNVERT*
.  .  .  .
.  .  .  '--MIN
.  .  .  .
.  .  .  '--OUT*
.  .  .  .
.  .  .  '--REMQUO+
.  .  .
```

G S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER
INVOCATION DIAGRAM OF THE DESIGN DIAGRAM GENERATOR (DDG)

```
.  .  '--GENERATE'CONTENTS'HEADER
.  .  .  .
.  .  .  '--ACCESS'OUT
.  .  .  .  .
.  .  .  .  '--CAT
.  .  .  .  .  .
.  .  .  .  .  '--CNVERT'
.  .  .  .  .  .
.  .  .  .  .  '--OUT'
.  .  .  .  .
.  .  .  .  '--IFORMAT
.  .  .  .  .  .
.  .  .  .  .  '--CNVERT'
.  .  .  .  .  .
.  .  .  .  .  '--ENCODE+
.  .  .  .  .
.  .  .  .  '--OUT'
.  .  .  .  .
.  .  .  .  '--REMQUO+
.  .  .  .  .
.  .  .  .  '--TRANSFER'WRITE'OUT
.  .  .  .     .
.  .  .  .     '--CAT
.  .  .  .     .  .
.  .  .  .     .  '--CNVERT'
.  .  .  .     .  .
.  .  .  .     .  '--OUT'
.  .  .  .     .
.  .  .  .     '--IFORMAT
.  .  .  .     .  .
.  .  .  .     .  '--CNVERT'
.  .  .  .     .  .
.  .  .  .     .  '--ENCODE+
.  .  .  .     .
.  .  .  .     '--MAX
.  .  .  .     .
.  .  .  .     '--OUT'
.  .  .  .
.  .  .  '--ADVANCE'PAGE
.  .  .  .  .
.  .  .  .  '--CAT
.  .  .  .  .  .
.  .  .  .  .  '--CNVERT'
.  .  .  .  .  .
.  .  .  .  .  '--OUT'
.  .  .  .  .
.  .  .  .  '--IFORMAT
.  .  .  .  .  .
.  .  .  .  .  '--CNVERT'
.  .  .  .  .  .
.  .  .  .  .  '--ENCODE+
.  .  .  .  .
.  .  .  .  '--OUT'
.  .  .  .  .
.  .  .  .  '--REMQUO+
.  .  .  .
```

INVOCATION DIAGRAM OF THE DESIGN DIAGRAM GENERATOR (DDG)

```
.  .  .  .--CNVERT*
.  .  .  .
.  .  .  .--IFORMAT
.  .  .  .  .
.  .  .  .  .--CNVERT*
.  .  .  .  .
.  .  .  .  .--ENCODE+
.  .  .  .
.  .  .  .--LENGTH
.  .  .  .  .
.  .  .  .  .--CNVERT*
.  .  .  .
.  .  .  .--OUT*
.  .  .
.  .  .--IFORMAT
.  .  .  .
.  .  .  .--CNVERT*
.  .  .  .
.  .  .  .--ENCODE+
.  .  .
.  .  .--OUT*
.  .  .
.  .  .--SPACES*
.  .
.  .--CONNECT'BOXES
.  .  .
.  .  .--ACCESS3
.  .  .  .
.  .  .  .--CAT
.  .  .  .  .
.  .  .  .  .--CNVERT*
.  .  .  .  .
.  .  .  .  .--OUT*
.  .  .  .
.  .  .  .--IFORMAT
.  .  .  .  .
.  .  .  .  .--CNVERT*
.  .  .  .  .
.  .  .  .  .--ENCODE+
.  .  .  .
.  .  .  .--MAX
.  .  .  .
.  .  .  .--OUT*
.  .  .  .
.  .  .  .--REMQUO+
.  .  .  .
.  .  .  .--TRANSFER'WRITE3
.  .  .     .
.  .  .     .--CAT
.  .  .     .  .
.  .  .     .  .--CNVERT*
.  .  .     .  .
.  .  .     .  .--OUT*
.  .  .     .
.  .  .     .-  'ORMAT
```

```
C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER                                        PAGE 18
INVOCATION DIAGRAM OF THE DESIGN DIAGRAM GENERATOR (DDG)

.  .  .     .  *--CNVERT*
.  .  .     .  .
.  .  .     .  *--ENCODE+
.  .  .     .
.  .  .     *--MAX
.  .  .     .
.  .  .     *--OUT*
.  .  .
.  .  *--ACCESS'OUT
.  .  .  .
.  .  .  *--CAT
.  .  .  .  .
.  .  .  .  *--CNVERT*
.  .  .  .  .
.  .  .  .  *--OUT*
.  .  .  .
.  .  .  *--IFORMAT
.  .  .  .  .
.  .  .  .  *--CNVERT*
.  .  .  .  .
.  .  .  .  *--ENCODE+
.  .  .  .
.  .  .  *--OUT*
.  .  .  .
.  .  .  *--REMQUO+
.  .  .  .
.  .  .  *--TRANSFER'WRITE'OUT
.  .  .     .
.  .  .     *--CAT
.  .  .     .  .
.  .  .     .  *--CNVERT*
.  .  .     .  .
.  .  .     .  *--OUT*
.  .  .     .
.  .  .     *--IFORMAT
.  .  .     .  .
.  .  .     .  *--CNVERT*
.  .  .     .  .
.  .  .     .  *--ENCODE+
.  .  .     .
.  .  .     *--MAX
.  .  .     .
.  .  .     *--OUT*
.  .  .
.  .  *--CAT
.  .  .  .
.  .  .  *--CNVERT*
.  .  .  .
.  .  .  *--OUT*
.  .  .
.  .  *--DRAW'LINE
.  .  .  .
.  .  .  *--ACCESS'OUT
.  .  .  .  .
.  .  .  .  *--CAT
.  .  .  .  .  .
```

```
* * * * * *--CNVERT*
* * * * * *
* * * * * *--OUT*
* * * * *
* * * * *--IFORMAT
* * * * * *
* * * * * *--CNVERT*
* * * * * *
* * * * * *--ENCODE+
* * * * *
* * * * *--OUT*
* * * * *
* * * * *--REMQUO+
* * * * *
* * * * *--TRANSFER*WRITE*OUT
* * * *   *
* * * *   *--CAT
* * * *   * *
* * * *   * *--CNVERT*
* * * *   * *
* * * *   * *--OUT*
* * * *   *
* * * *   *--IFORMAT
* * * *   * *
* * * *   * *--CNVERT*
* * * *   * *
* * * *   * *--ENCODE+
* * * *   *
* * * *   *--MAX
* * * *   *
* * * *   *--OUT*
* * * *
* * * *--CAT
* * * * *
* * * * *--CNVERT*
* * * * *
* * * * *--OUT*
* * * *
* * * *--DASHES
* * * * *
* * * * *--ACCESS*OUT
* * * *   *
* * * *   *--CAT
* * * *   * *
* * * *   * *--CNVERT*
* * * *   * *
* * * *   * *--OUT*
* * * *   *
* * * *   *--IFORMAT
* * * *   * *
* * * *   * *--CNVERT*
* * * *   * *
* * * *   * *--ENCODE+
* * * *   *
* * * *   *--OUT*
* * * *   *
```

```
•  •  •  •     •--REMQUO+
•  •  •  •     •
•  •  •  •     •--TRANSFER'WRITE'OUT
•  •  •  •        •
•  •  •  •        •--CAT
•  •  •  •        •  •
•  •  •  •        •  •--CNVERT•
•  •  •  •        •  •
•  •  •  •        •  •--OUT•
•  •  •  •        •
•  •  •  •        •--IFORMAT
•  •  •  •        •  •
•  •  •  •        •  •--CNVERT•
•  •  •  •        •  •
•  •  •  •        •  •--ENCODE+
•  •  •  •        •
•  •  •  •        •--MAX
•  •  •  •        •
•  •  •  •        •--OUT•
•  •  •  •
•  •  •  •--IFORMAT
•  •  •  •  •
•  •  •  •  •--CNVERT•
•  •  •  •  •
•  •  •  •  •--ENCODE+
•  •  •  •
•  •  •  •--OUT•
•  •  •  •
•  •  •  •--OUTPUT'HEADER
•  •  •  •  •
•  •  •  •  •--ACCESS'OUT
•  •  •  •  •  •
•  •  •  •  •  •--CAT
•  •  •  •  •  •  •
•  •  •  •  •  •  •--CNVERT•
•  •  •  •  •  •  •
•  •  •  •  •  •  •--OUT•
•  •  •  •  •  •
•  •  •  •  •  •--IFORMAT
•  •  •  •  •  •  •
•  •  •  •  •  •  •--CNVERT•
•  •  •  •  •  •  •
•  •  •  •  •  •  •--ENCODE+
•  •  •  •  •  •
•  •  •  •  •  •--OUT•
•  •  •  •  •  •
•  •  •  •  •  •--REMQUO+
•  •  •  •  •  •
•  •  •  •  •  •--TRANSFER'WRITE'OUT
•  •  •  •  •     •
•  •  •  •  •     •--CAT
•  •  •  •  •     •  •
•  •  •  •  •     •  •--CNVERT•
•  •  •  •  •     •  •
•  •  •  •  •     •  •--OUT•
•  •  •  •  •     •
```

INVOCATION DIAGRAM OF THE DESIGN DIAGRAM GENERATOR (DDG)

```
.  .  .  .  .     *--IFORMAT
.  .  .  .  .     .
.  .  .  .  .     .  *--CNVERT*
.  .  .  .  .     .  .
.  .  .  .  .     .  *--ENCODE+
.  .  .  .  .     .
.  .  .  .  .     *--MAX
.  .  .  .  .     .
.  .  .  .  .     *--OUT*
.  .  .  .  .
.  .  .  .  *--CAT
.  .  .  .  .  .
.  .  .  .  .  *--CNVERT*
.  .  .  .  .  .
.  .  .  .  .  *--OUT*
.  .  .  .  .
.  .  .  .  *--CNVERT*
.  .  .  .  .
.  .  .  .  *--IFORMAT
.  .  .  .  .  .
.  .  .  .  .  *--CNVERT*
.  .  .  .  .  .
.  .  .  .  .  *--ENCODE+
.  .  .  .  .
.  .  .  .  *--LENGTH
.  .  .  .  .  .
.  .  .  .  .  *--CNVERT*
.  .  .  .  .
.  .  .  .  *--OUT*
.  .  .  .  .
.  .  .  .  *--SPACES*
.  .  .  .
.  .  .  *--REMQUO+
.  .  .
.  .  *--IFORMAT
.  .  .  .
.  .  .  *--CNVERT*
.  .  .  .
.  .  .  *--ENCODE+
.  .  .
.  .  *--LENGTH
.  .  .  .
.  .  .  *--CNVERT*
.  .  .
.  .  *--OUT*
.  .  .
.  .  *--PH2ERR
.  .  .  .
.  .  .  *--OUT*
.  .  .  .
.  .  .  *--TRANSFER*WRITE3
.  .  .  .  .
.  .  .  .  *--CAT
.  .  .  .  .  .
.  .  .  .  .  *--CNVERT*
.  .  .  .  .
.  .  .  .  .  *--OUT*
.  .  .  .  .
.  .  .  .  *--IFORMAT
.  .  .  .  .  .
.  .  .  .  .  *--CNVERT*
.  .  .  .  .  .
.  .  .  .  .  *--ENCODE+
.  .  .  .  .
.  .  .  .  *--MAX
.  .  .  .  .
.  .  .  .  *--OUT*
.  .  .  .
.  .  .  *--TRANSFER'WRITE+
.  .  .     .
.  .  .     *--CAT
.  .  .     .  .
.  .  .     .  *--CNVERT*
.  .  .     .  .
.  .  .     .  *--OUT*
.  .  .     .
.  .  .     *--IFORMAT
.  .  .     .  .
.  .  .     .  *--CNVERT*
.  .  .     .  .
.  .  .     .  *--ENCODE+
.  .  .     .
.  .  .     *--MAX
.  .  .     .
.  .  .     *--OUT*
.  .  .
.  .  *--REMQUO+
.  .  .
.  .  *--STARS
.  .     .
.  .     *--ACCESS'OUT
.  .        .
.  .        *--CAT
.  .        .  .
.  .        .  *--CNVERT*
.  .        .  .
.  .        .  *--OUT*
.  .        .
.  .        *--IFORMAT
.  .        .  .
.  .        .  *--CNVERT*
.  .        .  .
.  .        .  *--ENCODE+
.  .        .
.  .        *--OUT*
.  .        .
.  .        *--REMQUO+
.  .        .
.  .        *--TRANSFER'WRITE'OUT
.  .           .
.  .           *--CAT
.  .           .  .
```

```
        *  *           *  *--CNVERT*
        *  *           *  *
        *  *           *  *--OUT*
        *  *           *
        *  *           *--IFORMAT
        *  *           *  *
        *  *           *  *--CNVERT*
        *  *           *  *
        *  *           *  *--ENCODE+
        *  *           *
        *  *           *--MAX
        *  *           *
        *  *           *--OUT*
        *  *
        *  *--CONSTRUCT'LINE
        *  *  *
        *  *  *--ACCESS'OUT
        *  *  *  *
        *  *  *  *--CAT
        *  *  *  *  *
        *  *  *  *  *--CNVERT*
        *  *  *  *  *
        *  *  *  *  *--OUT*
        *  *  *  *
        *  *  *  *--IFORMAT
        *  *  *  *  *
        *  *  *  *  *--CNVERT*
        *  *  *  *  *
        *  *  *  *  *--ENCODE+
        *  *  *  *
        *  *  *  *--OUT*
        *  *  *  *
        *  *  *  *--REMQUJ+
        *  *  *  *
        *  *  *  *--TRANSFER'WRITE'OUT
        *  *  *     *
        *  *  *     *--CAT
        *  *  *     *  *
        *  *  *     *  *--CNVERT*
        *  *  *     *  *
        *  *  *     *  *--OUT*
        *  *  *     *
        *  *  *     *--IFORMAT
        *  *  *     *  *
        *  *  *     *  *--CNVERT*
        *  *  *     *  *
        *  *  *     *  *--ENCODE+
        *  *  *     *
        *  *  *     *--MAX
        *  *  *     *
        *  *  *     *--OUT*
        *  *  *
        *  *  *--DASHES
        *  *  *  *
        *  *  *  *--ACCESS'OUT
        *  *  *     *
```

```
 .  .  .     .--CAT
 .  .  .     .  .
 .  .  .     .  .--CNVERT*
 .  .  .     .  .
 .  .  .     .  .--OUT*
 .  .  .     .
 .  .  .     .--IFORMAT
 .  .  .     .  .
 .  .  .     .  .--CNVERT*
 .  .  .     .  .
 .  .  .     .  .--ENCODE+
 .  .  .     .
 .  .  .     .--OUT*
 .  .  .     .
 .  .  .     .--REMQUO+
 .  .  .     .
 .  .  .     .--TRANSFER'WRITE'OUT
 .  .  .        .
 .  .  .        .--CAT'
 .  .  .        .  .
 .  .  .        .  .--CNVERT*
 .  .  .        .  .
 .  .  .        .  .--OUT*
 .  .  .        .
 .  .  .        .--IFORMAT
 .  .  .        .  .
 .  .  .        .  .--CNVERT*
 .  .  .        .  .
 .  .  .        .  .--ENCODE+
 .  .  .        .
 .  .  .        .--MAX
 .  .  .        .
 .  .  .        .--OUT*
 .  .  .
 .  .  .--LENGTH
 .  .  .  .
 .  .  .  .--CNVERT*
 .  .  .
 .  .  .--MIN
 .  .  .
 .  .  .--OUT*
 .  .  .
 .  .  .--OUTPUT'HEADER
 .  .  .  .
 .  .  .  .--ACCESS'OUT
 .  .  .  .  .
 .  .  .  .  .--CAT
 .  .  .  .  .  .
 .  .  .  .  .  .--CNVERT*
 .  .  .  .  .  .
 .  .  .  .  .  .--OUT*
 .  .  .  .  .
 .  .  .  .  .--IFORMAT
 .  .  .  .  .  .
 .  .  .  .  .  .--CNVERT*
 .  .  .  .  .  .
```

```
.  .  .  .  .  *--ENCODE+
.  .  .  .  .
.  .  .  .  *--OUT*
.  .  .  .  .
.  .  .  .  *--REMQUO+
.  .  .  .  .
.  .  .  .  *--TRANSFER*WRITE*OUT
.  .  .  .     .
.  .  .  .     *--CAT
.  .  .  .     .  .
.  .  .  .     .  *--CNVERT*
.  .  .  .     .  .
.  .  .  .     .  *--OUT*
.  .  .  .     .
.  .  .  .     *--IFORMAT
.  .  .  .     .  .
.  .  .  .     .  *--CNVERT*
.  .  .  .     .  .
.  .  .  .     .  *--ENCODE+
.  .  .  .     .
.  .  .  .     *--MAX
.  .  .  .     .
.  .  .  .     *--OUT*
.  .  .  .
.  .  .  *--CAT
.  .  .  .  .
.  .  .  .  *--CNVERT*
.  .  .  .  .
.  .  .  .  *--OUT*
.  .  .  .
.  .  .  *--CNVERT*
.  .  .  .
.  .  .  *--IFORMAT
.  .  .  .  .
.  .  .  .  *--CNVERT*
.  .  .  .  .
.  .  .  .  *--ENCODE+
.  .  .  .
.  .  .  *--LENGTH
.  .  .  .  .
.  .  .  .  *--CVVERT*
.  .  .  .
.  .  .  *--OUT*
.  .  .  .
.  .  .  *--SPACES*
.  .  .
.  .  *--REMQUO+
.  .
.  *--DISPLAY*STUMP*REF
.  .  .
.  .  *--ACCESS*OUT
.  .  .  .
.  .  .  *--CAT
.  .  .  .  .
.  .  .  .  *--CNVERT*
.  .  .  .  .
.  .  .  .  *--OUT*
.  .  .  .
.  .  .  *--IFORMAT
.  .  .  .  .
.  .  .  .  *--CNVERT*
.  .  .  .  .
.  .  .  .  *--ENCODE+
.  .  .  .
.  .  .  *--OUT*
.  .  .  .
.  .  .  *--REMQUO+
.  .  .  .
.  .  .  *--TRANSFER*WRITE*OUT
.  .  .     .
.  .  .     *--CAT
.  .  .     .  .
.  .  .     .  *--CNVERT*
.  .  .     .  .
.  .  .     .  *--OUT*
.  .  .     .
.  .  .     *--IFORMAT
.  .  .     .  .
.  .  .     .  *--CNVERT*
.  .  .     .  .
.  .  .     .  *--ENCODE+
.  .  .     .
.  .  .     *--MAX
.  .  .     .
.  .  .     *--OUT*
.  .  .
.  .  *--CAT
.  .  .  .
.  .  .  *--CNVERT*
.  .  .  .
.  .  .  *--OUT*
.  .  .
.  .  *--DASHES
.  .  .  .
.  .  .  *--ACCESS*OUT
.  .  .     .
.  .  .     *--CAT
.  .  .     .  .
.  .  .     .  *--CNVERT*
.  .  .     .  .
.  .  .     .  *--OUT*
.  .  .     .
.  .  .     *--IFORMAT
.  .  .     .  .
.  .  .     .  *--CNVERT*
.  .  .     .  .
.  .  .     .  *--ENCODE+
.  .  .     .
.  .  .     *--OUT*
.  .  .     .
.  .  .     *--REMQUO+
.  .  .     .
```

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER
INVOCATION DIAGRAM OF THE DESIGN DIAGRAM GENERATOR (DDG)

```
.  .  .     *--TRANSFER'WRITE'OUT
.  .  .        .
.  .  .        *--CAT
.  .  .        .  .
.  .  .        .  *--CNVERT*
.  .  .        .  .
.  .  .        .  *--OUT*
.  .  .        .
.  .  .        *--IFORMAT
.  .  .        .  .
.  .  .        .  *--CNVERT*
.  .  .        .  .
.  .  .        .  *--ENCODE+
.  .  .        .
.  .  .        *--MAX
.  .  .        .
.  .  .        *--OUT*
.  .  .
.  .  *--DRAW'LINE
.  .  .  .
.  .  .  *--ACCESS'OUT
.  .  .  .  .
.  .  .  .  *--CAT
.  .  .  .  .  .
.  .  .  .  .  *--CNVERT*
.  .  .  .  .  .
.  .  .  .  .  *--OUT*
.  .  .  .  .
.  .  .  .  *--IFORMAT
.  .  .  .  .  .
.  .  .  .  .  *--CNVERT*
.  .  .  .  .  .
.  .  .  .  .  *--ENCODE+
.  .  .  .  .
.  .  .  .  *--OUT*
.  .  .  .  .
.  .  .  .  *--REMQUO+
.  .  .  .  .
.  .  .  .  *--TRANSFER'WRITE'OUT
.  .  .  .     .
.  .  .  .     *--CAT
.  .  .  .     .  .
.  .  .  .     .  *--CNVERT*
.  .  .  .     .  .
.  .  .  .     .  *--OUT*
.  .  .  .     .
.  .  .  .     *--IFORMAT
.  .  .  .     .  .
.  .  .  .     .  *--CNVERT*
.  .  .  .     .  .
.  .  .  .     .  *--ENCODE+
.  .  .  .     .
.  .  .  .     *--MAX
.  .  .  .     .
.  .  .  .     *--OUT*
.  .  .  .
.  .  .  *--CAT
.  .  .  .  .
.  .  .  .  *--CNVERT*
.  .  .  .  .
.  .  .  .  *--OUT*
.  .  .  .
.  .  .  *--DASHES
.  .  .  .  .
.  .  .  .  *--ACCESS'OUT
.  .  .  .     .
.  .  .  .     *--CAT
.  .  .  .     .  .
.  .  .  .     .  *--CNVERT*
.  .  .  .     .  .
.  .  .  .     .  *--OUT*
.  .  .  .     .
.  .  .  .     *--IFORMAT
.  .  .  .     .  .
.  .  .  .     .  *--CNVERT*
.  .  .  .     .  .
.  .  .  .     .  *--ENCODE+
.  .  .  .     .
.  .  .  .     *--OUT*
.  .  .  .     .
.  .  .  .     *--REMQUO+
.  .  .  .     .
.  .  .  .     *--TRANSFER'WRITE'OUT
.  .  .  .        .
.  .  .  .        *--CAT
.  .  .  .        .  .
.  .  .  .        .  *--CNVERT*
.  .  .  .        .  .
.  .  .  .        .  *--OUT*
.  .  .  .        .
.  .  .  .        *--IFORMAT
.  .  .  .        .  .
.  .  .  .        .  *--CNVERT*
.  .  .  .        .  .
.  .  .  .        .  *--ENCODE+
.  .  .  .        .
.  .  .  .        *--MAX
.  .  .  .        .
.  .  .  .        *--OUT*
.  .  .  .
.  .  .  *--IFORMAT
.  .  .  .  .
.  .  .  .  *--CNVERT*
.  .  .  .  .
.  .  .  .  *--ENCODE+
.  .  .  .
.  .  .  *--OUT*
.  .  .  .
.  .  .  *--OUTPUT'HEADER
.  .  .  .  .
.  .  .  .  *--ACCESS'OUT
.  .  .  .     .
.  .  .  .  .  .--CAT
.  .  .  .  .  .  .
.  .  .  .  .  .  .--CNVERT*
.  .  .  .  .  .  .
.  .  .  .  .  .  .--OUT*
.  .  .  .  .  .
.  .  .  .  .  .--IFORMAT
.  .  .  .  .  .  .
.  .  .  .  .  .  .--CNVERT*
.  .  .  .  .  .  .
.  .  .  .  .  .  .--ENCODE+
.  .  .  .  .  .
.  .  .  .  .  .--OUT*
.  .  .  .  .  .
.  .  .  .  .  .--REMQUO+
.  .  .  .  .  .
.  .  .  .  .  .--TRANSFER'WRITE'OUT
.  .  .  .  .     .
.  .  .  .  .     .--CAT
.  .  .  .  .     .  .
.  .  .  .  .     .  .--CNVERT*
.  .  .  .  .     .  .
.  .  .  .  .     .  .--OUT*
.  .  .  .  .     .
.  .  .  .  .     .--IFORMAT
.  .  .  .  .     .  .
.  .  .  .  .     .  .--CNVERT*
.  .  .  .  .     .  .
.  .  .  .  .     .  .--ENCODE+
.  .  .  .  .     .
.  .  .  .  .     .--MAX
.  .  .  .  .     .
.  .  .  .  .     .--OUT*
.  .  .  .  .
.  .  .  .  .--CAT
.  .  .  .  .  .
.  .  .  .  .  .--CNVERT*
.  .  .  .  .  .
.  .  .  .  .  .--OUT*
.  .  .  .  .
.  .  .  .  .--CNVERT*
.  .  .  .  .
.  .  .  .  .--IFORMAT
.  .  .  .  .  .
.  .  .  .  .  .--CNVERT*
.  .  .  .  .  .
.  .  .  .  .  .--ENCODE+
.  .  .  .  .
.  .  .  .  .--LENGTH
.  .  .  .  .  .
.  .  .  .  .  .--CNVERT*
.  .  .  .  .
.  .  .  .  .--OUT*
.  .  .  .  .
.  .  .  .  .--SPACES*
.  .  .  .
.  .  .  *--REMQU3+
.  .  .
.  .  *--IFORMAT
.  .  .  .
.  .  .  *--CNVERT*
.  .  .  .
.  .  .  *--ENCODE+
.  .  .
.  .  *--LENGTH
.  .  .  .
.  .  .  *--CNVERT*
.  .  .
.  .  *--OUT*
.  .  .
.  .  *--PH2ERR
.  .  .  .
.  .  .  *--OUT*
.  .  .  .
.  .  .  *--TRANSFER*WRITE3
.  .  .  .  .
.  .  .  .  *--CAT
.  .  .  .  .  .
.  .  .  .  .  *--CNVERT*
.  .  .  .  .  .
.  .  .  .  .  *--OUT*
.  .  .  .  .
.  .  .  .  *--IFORMAT
.  .  .  .  .  .
.  .  .  .  .  *--CNVERT*
.  .  .  .  .  .
.  .  .  .  .  *--ENCODE+
.  .  .  .  .
.  .  .  .  *--MAX
.  .  .  .  .
.  .  .  .  *--OJT*
.  .  .  .
.  .  .  *--TRANSFER*WRITE4
.  .  .     .
.  .  .     *--CAT
.  .  .     .  .
.  .  .     .  *--CNVERT*
.  .  .     .  .
.  .  .     .  *--OUT*
.  .  .     .
.  .  .     *--IFORMAT
.  .  .     .  .
.  .  .     .  *--CNVERT*
.  .  .     .  .
.  .  .     .  *--ENCODE+
.  .  .     .
.  .  .     *--MAX
.  .  .     .
.  .  .     *--OUT*
.  .  .
.  .  *--REMQUO+
.  .  .
.  .  *--STARS
.  .     .
.  .     *--ACCESS*OUT
.  .        .
.  .        *--CAT
.  .        .  .
.  .        .  *--CNVERT*
.  .        .  .
.  .        .  *--OUT*
.  .        .
.  .        *--IFORMAT
.  .        .  .
.  .        .  *--CNVERT*
.  .        .  .
.  .        .  *--ENCODE+
.  .        .
.  .        *--OUT*
.  .        .
.  .        *--REMQUO+
.  .        .
.  .        *--TRANSFER*WRITE*OUT
.  .           .
.  .           *--CAT
.  .           .  .
.  .           .  *--CNVERT*
.  .           .  .
.  .           .  *--OUT*
.  .           .
.  .           *--IFORMAT
.  .           .  .
.  .           .  *--CNVERT*
.  .           .  .
.  .           .  *--ENCODE+
.  .           .
.  .           *--MAX
.  .           .
.  .           *--OUT*
.  .
.  *--EXTRACT*TEXT
.  .  .
.  .  *--ACCESS2
.  .  .  .
.  .  .  *--CAT
.  .  .  .  .
.  .  .  .  *--CNVERT*
.  .  .  .  .
.  .  .  .  *--OUT*
.  .  .  .
.  .  .  *--IFORMAT
.  .  .  .  .
.  .  .  .  *--CNVERT*
.  .  .  .  .
.  .  .  .  *--ENCODE+
.  .  .  .
.  .  .  *--OUT*
.  .  .  .
.  .  .  .--REMQUO+
.  .  .
.  .  .--ACCESS4
.  .  .  .
.  .  .  .--CAT
.  .  .  .  .
.  .  .  .  .--CVVERT*
.  .  .  .  .
.  .  .  .  .--OUT*
.  .  .  .
.  .  .  .--IFORMAT
.  .  .  .  .
.  .  .  .  .--CNVERT*
.  .  .  .  .
.  .  .  .  .--ENCODE+
.  .  .  .
.  .  .  .--MAX
.  .  .  .
.  .  .  .--OUT*
.  .  .  .
.  .  .  .--REMQUO+
.  .  .  .
.  .  .  .--TRANSFER.WRITE4
.  .  .     .
.  .  .     .--CAT
.  .  .     .  .
.  .  .     .  .--CNVERT*
.  .  .     .  .
.  .  .     .  .--OUT*
.  .  .     .
.  .  .     .--IFORMAT
.  .  .     .  .
.  .  .     .  .--CNVERT*
.  .  .     .  .
.  .  .     .  .--ENCODE+
.  .  .     .
.  .  .     .--MAX
.  .  .     .
.  .  .     .--OUT*
.  .  .
.  .  .--CAT
.  .  .  .
.  .  .  .--CNVERT*
.  .  .  .
.  .  .  .--OUT*
.  .  .
.  .  .--IFORMAT
.  .  .  .
.  .  .  .--CNVERT*
.  .  .  .
.  .  .  .--ENCODE+
.  .  .
.  .  .--LENGTH
.  .  .  .
.  .  .  .--CNVERT*
.  .  .
```

```
*  *  *--OUT*
*  *  *
*  *  *--SUBSTR
*  *     *
*  *     *--CAT
*  *     *  *
*  *     *  *--CNVERT*
*  *     *  *
*  *     *  *--OUT*
*  *     *
*  *     *--CNVERT*
*  *     *
*  *     *--ENCODE+
*  *     *
*  *     *--OUT*
*  *
*  *--IFORMAT
*  *  *
*  *  *--CNVERT*
*  *  *
*  *  *--ENCODE+
*  *
*  *--INIT*BLOCK*CONSTANTS
*  *  *
*  *  *--ACCESS3
*  *     *
*  *     *--CAT
*  *     *  *
*  *     *  *--CNVERT*
*  *     *  *
*  *     *  *--OUT*
*  *     *
*  *     *--IFORMAT
*  *     *  *
*  *     *  *--CNVERT*
*  *     *  *
*  *     *  *--ENCODE+
*  *     *
*  *     *--MAX
*  *     *
*  *     *--OUT*
*  *     *
*  *     *--REMQUO+
*  *     *
*  *     *--TRANSFER*WRITE3
*  *        *
*  *        *--CAT
*  *        *  *
*  *        *  *--CNVERT*
*  *        *  *
*  *        *  *--OUT*
*  *        *
*  *        *--IFORMAT
*  *        *  *
*  *        *  *--CNVERT*
```

INVOCATION DIAGRAM OF THE DESIGN DIAGRAM GENERATOR (DDG)

```
.  .        .  '--ENCODE+
.  .        .
.  .        '--MAX
.  .        .
.  .        '--OUT'
.  .
.  '--LENGTH
.  .  .
.  .  '--CNVERT'
.  .
.  '--OUT'
.  .
.  '--OUTPUT'BOX'BOTTOM
.  .  .
.  .  '--ACCESS3
.  .  .  .
.  .  .  '--CAT
.  .  .  .  .
.  .  .  .  '--CNVERT'
.  .  .  .  .
.  .  .  .  '--OUT'
.  .  .  .
.  .  .  '--IFORMAT
.  .  .  .  .
.  .  .  .  '--CNVERT'
.  .  .  .  .
.  .  .  .  '--ENCODE+
.  .  .  .
.  .  .  '--MAX
.  .  .  .
.  .  .  '--OUT'
.  .  .  .
.  .  .  '--REMQUO+
.  .  .  .
.  .  .  '--TRANSFER'WRITE3
.  .  .     .
.  .  .     '--CAT
.  .  .     .  .
.  .  .     .  '--CNVERT'
.  .  .     .  .
.  .  .     .  '--OUT'
.  .  .     .
.  .  .     '--IFORMAT
.  .  .     .  .
.  .  .     .  '--CNVERT'
.  .  .     .  .
.  .  .     .  '--ENCODE+
.  .  .     .
.  .  .     '--MAX
.  .  .     .
.  .  .     '--OUT'
.  .  .
.  .  '--ACCESS'OUT
.  .  .  .
.  .  .  '--CAT
.  .  .  .  .
```

```
*  *  *  *  *--CNVERT*
*  *  *  *  *
*  *  *  *  *--OUT*
*  *  *  *
*  *  *  *--IFORMAT
*  *  *  *  *
*  *  *  *  *--CNVERT*
*  *  *  *  *
*  *  *  *  *--ENCODE+
*  *  *  *
*  *  *  *--OUT*
*  *  *  *
*  *  *  *--REMQUO+
*  *  *  *
*  *  *  *--TRANSFER*WRITE*OUT
*  *  *     *
*  *  *     *--CAT
*  *  *     *  *
*  *  *     *  *--CNVERT*
*  *  *     *  *
*  *  *     *  *--OUT*
*  *  *     *
*  *  *     *--IFORMAT
*  *  *     *  *
*  *  *     *  *--CNVERT*
*  *  *     *  *
*  *  *     *  *--ENCODE+
*  *  *     *
*  *  *     *--MAX
*  *  *     *
*  *  *     *--OJT*
*  *  *
*  *  *--CAT
*  *  *  *
*  *  *  *--CNVERT*
*  *  *  *
*  *  *  *--OUT*
*  *  *
*  *  *--DRAW*LINE
*  *  *  *
*  *  *  *--ACCESS*OUT
*  *  *  *  *
*  *  *  *  *--CAT
*  *  *  *  *  *
*  *  *  *  *  *--CNVERT*
*  *  *  *  *  *
*  *  *  *  *  *--OUT*
*  *  *  *  *
*  *  *  *  *--IFORMAT
*  *  *  *  *  *
*  *  *  *  *  *--CNVERT*
*  *  *  *  *  *
*  *  *  *  *  *--ENCODE+
*  *  *  *  *
*  *  *  *  *--OUT*
*  *  *  *  *
```

```
.  .  .  .  .--REMQUO+
.  .  .  .  .
.  .  .  .  .--TRANSFER'WRITE'OUT
.  .  .  .     .
.  .  .  .     .--CAT
.  .  .  .     .  .
.  .  .  .     .  .--CNVERT*
.  .  .  .     .  .
.  .  .  .     .  .--OUT*
.  .  .  .     .
.  .  .  .     .--IFORMAT
.  .  .  .     .  .
.  .  .  .     .  .--CNVERT*
.  .  .  .     .  .
.  .  .  .     .  .--ENCODE+
.  .  .  .     .
.  .  .  .     .--MAX
.  .  .  .     .
.  .  .  .     .--OUT*
.  .  .  .
.  .  .  .--CAT
.  .  .  .  .
.  .  .  .  .--CNVERT*
.  .  .  .  .
.  .  .  .  .--OUT*
.  .  .  .
.  .  .  .--DASHES
.  .  .  .  .
.  .  .  .  .--ACCESS'OUT
.  .  .  .     .
.  .  .  .     .--CAT
.  .  .  .     .  .
.  .  .  .     .  .--CNVERT*
.  .  .  .     .  .
.  .  .  .     .  .--OUT*
.  .  .  .     .
.  .  .  .     .--IFORMAT
.  .  .  .     .  .
.  .  .  .     .  .--CNVERT*
.  .  .  .     .  .
.  .  .  .     .  .--ENCODE+
.  .  .  .     .
.  .  .  .     .--OUT*
.  .  .  .     .
.  .  .  .     .--REMQUO+
.  .  .  .     .
.  .  .  .     .--TRANSFER'WRITE'OUT
.  .  .  .        .
.  .  .  .        .--CAT
.  .  .  .        .  .
.  .  .  .        .  .--CNVERT*
.  .  .  .        .  .
.  .  .  .        .  .--OUT*
.  .  .  .        .
.  .  .  .        .--IFORMAT
.  .  .  .        .  .
.  .  .  .        .  *--CNVERT*
.  .  .  .        .  .
.  .  .  .        .  *--ENCODE+
.  .  .  .        .
.  .  .  .        *--MAX
.  .  .  .        .
.  .  .  .        *--OUT*
.  .  .  .
.  .  .  *--IFORMAT
.  .  .  .  .
.  .  .  .  *--CNVERT*
.  .  .  .  .
.  .  .  .  *--ENCODE+
.  .  .  .
.  .  .  *--OUT*
.  .  .  .
.  .  .  *--OUTPUT*HEADER
.  .  .  .  .
.  .  .  .  *--ACCESS*OUT
.  .  .  .  .  .
.  .  .  .  .  *--CAT
.  .  .  .  .  .  .
.  .  .  .  .  .  *--CNVERT*
.  .  .  .  .  .  .
.  .  .  .  .  .  *--OUT*
.  .  .  .  .  .
.  .  .  .  .  *--IFORMAT
.  .  .  .  .  .  .
.  .  .  .  .  .  *--CNVERT*
.  .  .  .  .  .  .
.  .  .  .  .  .  *--ENCODE+
.  .  .  .  .  .
.  .  .  .  .  *--OUT*
.  .  .  .  .  .
.  .  .  .  .  *--REMQUO+
.  .  .  .  .  .
.  .  .  .  .  *--TRANSFER*WRITE*OUT
.  .  .  .  .     .
.  .  .  .  .     *--CAT
.  .  .  .  .     .  .
.  .  .  .  .     .  *--CNVERT*
.  .  .  .  .     .  .
.  .  .  .  .     .  *--OUT*
.  .  .  .  .     .
.  .  .  .  .     *--IFORMAT
.  .  .  .  .     .  .
.  .  .  .  .     .  *--CNVERT*
.  .  .  .  .     .  .
.  .  .  .  .     .  *--ENCODE+
.  .  .  .  .     .
.  .  .  .  .     *--MAX
.  .  .  .  .     .
.  .  .  .  .     *--OUT*
.  .  .  .  .
.  .  .  .  *--CAT
.  .  .  .  .  '--CNVERT*
.  .  .  .  .  .
.  .  .  .  .  '--OUT*
.  .  .  .  .
.  .  .  .  '--CNVERT*
.  .  .  .  .
.  .  .  .  '--IFORMAT
.  .  .  .  .  .
.  .  .  .  .  '--CNVERT*
.  .  .  .  .  .
.  .  .  .  .  '--ENCODE+
.  .  .  .  .
.  .  .  .  '--LENGTH
.  .  .  .  .  .
.  .  .  .  .  '--CNVERT*
.  .  .  .  .
.  .  .  .  '--OUT*
.  .  .  .  .
.  .  .  .  '--SPACES*
.  .  .  .
.  .  .  '--REMQUO+
.  .  .
.  .  '--IFORMAT
.  .  .  .
.  .  .  '--CNVERT*
.  .  .  .
.  .  .  '--ENCODE+
.  .  .
.  .  '--MAX
.  .  .
.  .  '--OUT*
.  .  .
.  .  '--REMQUO+
.  .  .
.  .  '--STARS
.  .     .
.  .     '--ACCESS'OUT
.  .        .
.  .        '--CAT
.  .        .  .
.  .        .  '--CNVERT*
.  .        .  .
.  .        .  '--OUT*
.  .        .
.  .        '--IFORMAT
.  .        .  .
.  .        .  '--CNVERT*
.  .        .  .
.  .        .  '--ENCODE+
.  .        .
.  .        '--OUT*
.  .        .
.  .        '--REMQUO+
.  .        .
.  .        '--TRANSFER'WRITE'OUT
.  .           .
.  .           *--CAT
.  .           .  .
.  .           .  *--CNVERT*
.  .           .  .
.  .           .  *--OUT*
.  .           .
.  .           *--IFORMAT
.  .           .  .
.  .           .  *--CNVERT*
.  .           .  .
.  .           .  *--ENCODE+
.  .           .
.  .           *--MAX
.  .           .
.  .           *--OUT*
.  .
.  *--OUTPUT*BOX*TOP
.  .  .
.  .  *--ACCESS*OUT
.  .  .  .
.  .  .  *--CAT
.  .  .  .  .
.  .  .  .  *--CVVERT*
.  .  .  .  .
.  .  .  .  *--OJT*
.  .  .  .
.  .  .  *--IFORMAT
.  .  .  .  .
.  .  .  .  *--CNVERT*
.  .  .  .  .
.  .  .  .  *--ENCODE+
.  .  .  .
.  .  .  *--OUT*
.  .  .  .
.  .  .  *--REMQUD+
.  .  .  .
.  .  .  *--TRANSFER*WRITE*OUT
.  .  .     .
.  .  .     *--CAT
.  .  .     .  .
.  .  .     .  *--CNVERT*
.  .  .     .  .
.  .  .     .  *--OUT*
.  .  .     .
.  .  .     *--IFORMAT
.  .  .     .  .
.  .  .     .  *--CNVERT*
.  .  .     .  .
.  .  .     .  *--ENCODE+
.  .  .     .
.  .  .     *--MAX
.  .  .     .
.  .  .     *--OJT*
.  .  .
.  .  *--OUT*
.  .  .
```

```
*  *  *--STARS
*  *     *
*  *     *--ACCESS*OUT
*  *        *
*  *        *--CAT
*  *        *  *
*  *        *  *--CNVERT*
*  *        *  *
*  *        *  *--OUT*
*  *        *
*  *        *--IFORMAT
*  *        *  *
*  *        *  *--CNVERT*
*  *        *  *
*  *        *  *--ENCODE+
*  *        *
*  *        *--OUT*
*  *        *
*  *        *--REMQUO+
*  *        *
*  *        *--TRANSFER*WRITE*OUT
*  *           *
*  *           *--CAT
*  *           *  *
*  *           *  *--CNVERT*
*  *           *  *
*  *           *  *--OUT*
*  *           *
*  *           *--IFORMAT
*  *           *  *
*  *           *  *--CNVERT*
*  *           *  *
*  *           *  *--ENCODE+
*  *           *
*  *           *--MAX
*  *           *
*  *           *--OUT*
*  *
*  *--OUTPUT*HEADER
*  *  *
*  *  *--ACCESS*OUT
*  *  *  *
*  *  *  *--CAT
*  *  *  *  *
*  *  *  *  *--CNVERT*
*  *  *  *  *
*  *  *  *  *--OUT*
*  *  *  *
*  *  *  *--IFORMAT
*  *  *  *  *
*  *  *  *  *--CNVERT*
*  *  *  *  *
*  *  *  *  *--ENCODE+
*  *  *  *
*  *  *  *--OUT*
```

```
C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER                                       PAGE 33
INVOCATION DIAGRAM OF THE DESIGN DIAGRAM GENERATOR (DDG)

      .  .  .  '--RENQUO+
      .  .  .  .
      .  .  .  '--TRANSFER'WRITE'OUT
      .  .  .     .
      .  .  .     '--CAT
      .  .  .     .  .
      .  .  .     .  '--CNVERT'
      .  .  .     .  .
      .  .  .     .  '--OUT'
      .  .  .     .
      .  .  .     '--IFORMAT
      .  .  .     .  .
      .  .  .     .  '--CNVERT'
      .  .  .     .  .
      .  .  .     .  '--ENCODE+
      .  .  .     .
      .  .  .     '--MAX
      .  .  .     .
      .  .  .     '--OUT'
      .  .  .
      .  .  '--CAT
      .  .  .  .
      .  .  .  '--CNVERT'
      .  .  .  .
      .  .  .  '--OUT'
      .  .  .
      .  .  '--CNVERT'
      .  .  .
      .  .  '--IFORMAT
      .  .  .  .
      .  .  .  '--CNVERT'
      .  .  .  .
      .  .  .  '--ENCODE+
      .  .  .
      .  .  '--LENGTH
      .  .  .  .
      .  .  .  '--CNVERT'
      .  .  .
      .  .  '--OUT'
      .  .  .
      .  .  '--SPACES'
      .  .
      .  '--OUTPUT'TITLE'PAGE
      .  .  .
      .  .  '--ACCESS'OUT
      .  .  .  .
      .  .  .  '--CAT
      .  .  .  .  .
      .  .  .  .  '--CNVERT'
      .  .  .  .  .
      .  .  .  .  '--OUT'
      .  .  .  .
      .  .  .  '--IFORMAT
      .  .  .  .  .
      .  .  .  .  '--CNVERT'
.  .  .  .  *--ENCODE+
.  .  .  .
.  .  .  *--OUT*
.  .  .  .
.  .  .  *--REMQUO+
.  .  .  .
.  .  .  *--TRANSFER*WRITE*OUT
.  .  .     .
.  .  .     *--CAT
.  .  .     .  .
.  .  .     .  *--CNVERT*
.  .  .     .  .
.  .  .     .  *--OUT*
.  .  .     .
.  .  .     *--IFORMAT
.  .  .     .  .
.  .  .     .  *--CNVERT*
.  .  .     .  .
.  .  .     .  *--ENCJDE+
.  .  .     .
.  .  .     *--MAX
.  .  .     .
.  .  .     *--OJT*
.  .  .
.  .  *--IFORMAT
.  .  .  .
.  .  .  *--CNVERT*
.  .  .  .
.  .  .  *--ENCODE+
.  .  .
.  .  *--LENGTH
.  .  .  .
.  .  .  *--CNVERT*
.  .  .
.  .  *--REMQUO+
.  .
.  *--PART2*INIT
.  .  .
.  .  *--CNVERT*
.  .  .
.  .  *--LENGTH
.  .  .  .
.  .  .  *--CNVERT*
.  .
.  *--REMQUO+
.  .
.  *--SPACES*
.
*--PH2ERR
.  .
.  *--OUT*
.  .
.  *--TRANSFER*WRITE3
.  .  .
.  .  *--CAT
.  .  .  .
.  .  .  .--CNVERT*
.  .  .  .
.  .  .  .--OUT*
.  .  .
.  .  .--IFORMAT
.  .  .  .
.  .  .  .--CNVERT*
.  .  .  .
.  .  .  .--ENCODE+
.  .  .
.  .  .--MAX
.  .  .
.  .  .--OUT*
.  .
.  .--TRANSFER'WRITE4
.     .
.     .--CAT
.     .  .
.     .  .--CNVERT*
.     .  .
.     .  .--OUT*
.     .
.     .--IFORMAT
.     .  .
.     .  .--CNVERT*
.     .  .
.     .  .--ENCODE+
.     .
.     .--MAX
.     .
.     .--OUT*
.
.--UPDATE'FILE3
   .
   .--ACCESS3
   .  .
   .  .--CAT
   .  .  .
   .  .  .--CNVERT*
   .  .  .
   .  .  .--OUT*
   .  .
   .  .--IFORMAT
   .  .  .
   .  .  .--CNVERT*
   .  .  .
   .  .  .--ENCODE+
   .  .
   .  .--MAX
   .  .
   .  .--OUT*
   .  .
   .  .--REMQUO+
   .  .
   .  .--TRANSFER'WRITE3
   .
```

```
.     *--CAT
.     .  .
.     .  *--CNVERT*
.     .  .
.     .  *--OJT*
.     .
.     *--IFORMAT
.     .  .
.     .  *--CNVERT*
.     .  .
.     .  *--ENCODE+
.     .
.     *--MAX
.     .
.     *--OUT*
.
*--BOX*MAP
.
*--CAT
.  .
.  *--CNVERT*
.  .
.  *--OUT*
.
*--CLOSE*REC
.  .
.  *--ACCESS3
.  .  .
.  .  *--CAT
.  .  .  .
.  .  .  *--CNVERT*
.  .  .  .
.  .  .  *--OUT*
.  .  .
.  .  *--IFORMAT
.  .  .  .
.  .  .  *--CNVERT*
.  .  .  .
.  .  .  *--ENCODE+
.  .  .
.  .  *--MAX
.  .  .
.  .  *--OUT*
.  .  .
.  .  *--REMQJO+
.  .  .
.  .  *--TRANSFER*WRITE3
.  .     .
.  .     *--CAT
.  .     .  .
.  .     .  *--CNVERT*
.  .     .  .
.  .     .  *--OUT*
.  .     .
.  .     *--IFORMAT
.  .     .
```

```
*  *     *  *--CNVERT*
*  *     *  *
*  *     *  *--ENCODE+
*  *     *
*  *     *--MAX
*  *     *
*  *     *--OUT*
*  *
*  *--PLACE
*  *  *
*  *  *--ACCESS3
*  *  *  *
*  *  *  *--CAT
*  *  *  *  *
*  *  *  *  *--CNVERT*
*  *  *  *  *
*  *  *  *  *--OUT*
*  *  *  *
*  *  *  *--IFORMAT
*  *  *  *  *
*  *  *  *  *--CNVERT*
*  *  *  *  *
*  *  *  *  *--ENCODE+
*  *  *  *
*  *  *  *--MAX
*  *  *  *
*  *  *  *--OUT*
*  *  *  *
*  *  *  *--RENQUO+
*  *  *  *
*  *  *  *--TRANSFER*WRITE3
*  *  *     *
*  *  *     *--CAT
*  *  *     *  *
*  *  *     *  *--CNVERT*
*  *  *     *  *
*  *  *     *  *--OUT*
*  *  *     *
*  *  *     *--IFORMAT
*  *  *     *  *
*  *  *     *  *--CNVERT*
*  *  *     *  *
*  *  *     *  *--ENCODE+
*  *  *     *
*  *  *     *--MAX
*  *  *     *
*  *  *     *--OUT*
*  *  *
*  *  *--CAT
*  *  *  *
*  *  *  *--CNVERT*
*  *  *  *
*  *  *  *--OUT*
*  *  *
*  *  *--IFORMAT
*  *  *  *
```

```
. . . *--CNVERT*
. . . .
. . . *--ENCODE+
. . .
. . *--INSERT
. . . .
. . . *--EXTRACT*NAME
. . . . .
. . . . *--ACCESS2
. . . . . .
. . . . . *--CAT
. . . . . . .
. . . . . . *--CNVERT*
. . . . . . .
. . . . . . *--OUT*
. . . . . .
. . . . . *--IFORMAT
. . . . . . .
. . . . . . *--CNVERT*
. . . . . . .
. . . . . . *--ENCODE+
. . . . . .
. . . . . *--OUT*
. . . . . .
. . . . . *--REMQUO+
. . . . .
. . . . *--CAT
. . . . . .
. . . . . *--CNVERT*
. . . . . .
. . . . . *--OUT*
. . . . .
. . . . *--CNVERT*
. . . . .
. . . . *--LENGTH
. . . . . .
. . . . . *--CNVERT*
. . . . .
. . . . *--OUT*
. . . . .
. . . . *--PH2ERR
. . . . . .
. . . . . *--OUT*
. . . . . .
. . . . . *--TRANSFER*WRITE3
. . . . . . .
. . . . . . *--CAT
. . . . . . . .
. . . . . . . *--CNVERT*
. . . . . . . .
. . . . . . . *--OUT*
. . . . . . .
. . . . . . *--IFORMAT
. . . . . . . .
. . . . . . . *--CNVERT*
. . . . . . . .
.  .  .  .  .  .  .  *--ENCODE+
.  .  .  .  .  .  .
.  .  .  .  .  .  *--MAX
.  .  .  .  .  .  .
.  .  .  .  .  .  *--OUT*
.  .  .  .  .  .
.  .  .  .  .  *--TRANSFER*WRITE4
.  .  .  .  .  .  .
.  .  .  .  .  .  *--CAT
.  .  .  .  .  .  .  .
.  .  .  .  .  .  .  *--CNVERT*
.  .  .  .  .  .  .  .
.  .  .  .  .  .  .  *--OUT*
.  .  .  .  .  .  .
.  .  .  .  .  .  *--IFORMAT
.  .  .  .  .  .  .  .
.  .  .  .  .  .  .  *--CNVERT*
.  .  .  .  .  .  .  .
.  .  .  .  .  .  .  *--ENCODE+
.  .  .  .  .  .  .
.  .  .  .  .  .  *--MAX
.  .  .  .  .  .  .
.  .  .  .  .  .  *--OUT*
.  .  .  .  .
.  .  .  .  *--SUBSTR
.  .  .  .  .  .
.  .  .  .  .  *--CAT
.  .  .  .  .  .  .
.  .  .  .  .  .  *--CNVERT*
.  .  .  .  .  .  .
.  .  .  .  .  .  *--OUT*
.  .  .  .  .  .
.  .  .  .  .  *--CNVERT*
.  .  .  .  .  .
.  .  .  .  .  *--ENCODE+
.  .  .  .  .  .
.  .  .  .  .  *--OUT*
.  .  .  .
.  .  .  *--PM2ERR
.  .  .  .  .
.  .  .  .  *--OUT*
.  .  .  .  .
.  .  .  .  *--TRANSFER*WRITE3
.  .  .  .  .  .
.  .  .  .  .  *--CAT
.  .  .  .  .  .  .
.  .  .  .  .  .  *--CNVERT*
.  .  .  .  .  .  .
.  .  .  .  .  .  *--OUT*
.  .  .  .  .  .
.  .  .  .  .  *--IFORMAT
.  .  .  .  .  .  .
.  .  .  .  .  .  *--CNVERT*
.  .  .  .  .  .  .
.  .  .  .  .  .  *--ENCODE+
.  .  .  .  .  .--MAX
.  .  .  .  .  .
.  .  .  .  .  .--OUT*
.  .  .  .  .
.  .  .  .  .--TRANSFER*WRITE4
.  .  .  .     .
.  .  .  .     .--CAT
.  .  .  .     .  .
.  .  .  .     .  .--CNVERT*
.  .  .  .     .  .
.  .  .  .     .  .--OUT*
.  .  .  .     .
.  .  .  .     .--IFORMAT
.  .  .  .     .  .
.  .  .  .     .  .--CNVERT*
.  .  .  .     .  .
.  .  .  .     .  .--ENCODE+
.  .  .  .     .
.  .  .  .     .--MAX
.  .  .  .     .
.  .  .  .     .--OUT*
.  .  .  .
.  .  .  .--SPACES*
.  .  .
.  .  .--MAX
.  .  .
.  .  .--MIN
.  .  .
.  .  .--OUT*
.  .  .
.  .  .--PH2ERR
.  .  .  .
.  .  .  .--OUT*
.  .  .  .
.  .  .  .--TRANSFER*WRITE3
.  .  .  .  .
.  .  .  .  .--CAT
.  .  .  .  .  .
.  .  .  .  .  .--CNVERT*
.  .  .  .  .  .
.  .  .  .  .  .--OUT*
.  .  .  .  .
.  .  .  .  .--IFORMAT
.  .  .  .  .  .
.  .  .  .  .  .--CNVERT*
.  .  .  .  .  .
.  .  .  .  .  .--ENCODE+
.  .  .  .  .
.  .  .  .  .--MAX
.  .  .  .  .
.  .  .  .  .--OUT*
.  .  .  .
.  .  .  .--TRANSFER*WRITE4
.  .  .     .
.  .  .     .--CAT
.  .  .     .  .
.  .  .     .  *--CNVERT*
.  .  .     .  .
.  .  .     .  *--OUT*
.  .  .     .
.  .  .     *--IFORMAT
.  .  .     .  .
.  .  .     .  *--CNVERT*
.  .  .     .  .
.  .  .     .  *--ENCODE+
.  .  .     .
.  .  .     *--MAX
.  .  .     .
.  .  .     *--OUT*
.  .  .
.  .  *--PUSH*LAYOUT*INFO
.  .  .  .
.  .  .  *--PH2ERR
.  .  .     .
.  .  .     *--OUT*
.  .  .     .
.  .  .     *--TRANSFER*WRITE3
.  .  .     .  .
.  .  .     .  *--CAT
.  .  .     .  .  .
.  .  .     .  .  *--CNVERT*
.  .  .     .  .  .
.  .  .     .  .  *--OUT*
.  .  .     .  .
.  .  .     .  *--IFORMAT
.  .  .     .  .  .
.  .  .     .  .  *--CNVERT*
.  .  .     .  .  .
.  .  .     .  .  *--ENCODE+
.  .  .     .  .
.  .  .     .  *--MAX
.  .  .     .  .
.  .  .     .  *--OUT*
.  .  .     .
.  .  .     *--TRANSFER*WRITE4
.  .  .        .
.  .  .        *--CAT
.  .  .        .  .
.  .  .        .  *--CNVERT*
.  .  .        .  .
.  .  .        .  *--OUT*
.  .  .        .
.  .  .        *--IFORMAT
.  .  .        .  .
.  .  .        .  *--CNVERT*
.  .  .        .  .
.  .  .        .  *--ENCODE+
.  .  .        .
.  .  .        *--MAX
.  .  .        .
.  .  .        *--OUT*
.  .  '--REMQUO+
.  .
.  '--REMQUO+
.  .
.  '--RESOLVE'STUMP
.     .
.     '--ACCESS3
.     .  .
.     .  '--CAT
.     .  .  .
.     .  .  '--CNVERT'
.     .  .  .
.     .  .  '--OUT'
.     .  .
.     .  '--IFORMAT
.     .  .  .
.     .  .  '--CNVERT'
.     .  .  .
.     .  .  '--ENCODE+
.     .  .
.     .  '--MAX
.     .  .
.     .  '--OUT'
.     .  .
.     .  '--REMQUO+
.     .  .
.     .  '--TRANSFER'WRITE3
.     .     .
.     .     '--CAT
.     .     .  .
.     .     .  '--CNVERT'
.     .     .  .
.     .     .  '--OUT'
.     .     .
.     .     '--IFORMAT
.     .     .  .
.     .     .  '--CNVERT'
.     .     .  .
.     .     .  '--ENCODE+
.     .     .
.     .     '--MAX
.     .     .
.     .     '--OUT'
.     .
.     '--CAT
.     .  .
.     .  '--CNVERT'
.     .  .
.     .  '--OUT'
.     .
.     '--IFORMAT
.     .  .
.     .  '--CNVERT'
.     .  .
.     .  '--ENCODE+
.     .
.       .--LEGAL'STUMP
.       .
.       .--OUT*
.       .
.       .--PH2ERR
.       .   .
.       .   .--OUT*
.       .   .
.       .   .--TRANSFER'WRITE3
.       .   .   .
.       .   .   .--CAT
.       .   .   .   .
.       .   .   .   .--CNVERT*
.       .   .   .   .
.       .   .   .   .--OUT*
.       .   .   .
.       .   .   .--IFORMAT
.       .   .   .   .
.       .   .   .   .--CNVERT*
.       .   .   .   .
.       .   .   .   .--ENCODE+
.       .   .   .
.       .   .   .--MAX
.       .   .   .
.       .   .   .--OUT*
.       .   .
.       .   .--TRANSFER'WRITE4
.       .       .
.       .       .--CAT
.       .       .   .
.       .       .   .--CNVERT*
.       .       .   .
.       .       .   .--OUT*
.       .       .
.       .       .--IFORMAT
.       .       .   .
.       .       .   .--CNVERT*
.       .       .   .
.       .       .   .--ENCODE+
.       .       .
.       .       .--MAX
.       .       .
.       .       .--OUT*
.       .
.       .--PLACE
.           .
.           .--ACCESS3
.           .   .
.           .   .--CAT
.           .   .   .
.           .   .   .--CNVERT*
.           .   .   .
.           .   .   .--OUT*
.           .   .
.           .   .--IFORMAT
.           .   .
.         .  .  *--CNVERT*
.         .  .  .
.         .  .  *--ENCODE+
.         .  .
.         .  *--MAX
.         .  .
.         .  *--OUT*
.         .  .
.         .  *--REMQUO+
.         .  .
.         .  *--TRANSFER*WRITE3
.         .     .
.         .     *--CAT
.         .     .  .
.         .     .  *--CNVERT*
.         .     .  .
.         .     .  *--OUT*
.         .     .
.         .     *--IFORMAT
.         .     .  .
.         .     .  *--CNVERT*
.         .     .  .
.         .     .  *--ENCODE+
.         .     .
.         .     *--MAX
.         .     .
.         .     *--OUT*
.         .
.         *--CAT
.         .  .
.         .  *--CNVERT*
.         .  .
.         .  *--OUT*
.         .
.         *--IFORMAT
.         .  .
.         .  *--CNVERT*
.         .  .
.         .  *--ENCODE+
.         .
.         *--INSERT
.         .  .
.         .  *--EXTRACT*NAME
.         .  .  .
.         .  .  *--ACCESS2
.         .  .  .  .
.         .  .  .  *--CAT
.         .  .  .  .  .
.         .  .  .  .  *--CNVERT*
.         .  .  .  .  .
.         .  .  .  .  *--OUT*
.         .  .  .  .
.         .  .  .  *--IFORMAT
.         .  .  .  .  .
.         .  .  .  .  *--CNVERT*
.         .  .  .  .  .
.     . . . . *--ENCODE+
.     . . . .
.     . . . *--OUT*
.     . . . .
.     . . . *--REMQUO+
.     . . .
.     . . *--CAT
.     . . . .
.     . . . *--CNVERT*
.     . . . .
.     . . . *--OUT*
.     . . .
.     . . *--CNVERT*
.     . . .
.     . . *--LENGTH
.     . . . .
.     . . . *--CNVERT*
.     . . .
.     . . *--OUT*
.     . . .
.     . . *--PH2ERR
.     . . . .
.     . . . *--OUT*
.     . . . .
.     . . . *--TRANSFER'WRITE3
.     . . . . .
.     . . . . *--CAT
.     . . . . . .
.     . . . . . *--CNVERT*
.     . . . . . .
.     . . . . . *--OUT*
.     . . . . .
.     . . . . *--IFORMAT
.     . . . . . .
.     . . . . . *--CNVERT*
.     . . . . . .
.     . . . . . *--ENCODE+
.     . . . . .
.     . . . . *--MAX
.     . . . . .
.     . . . . *--OUT*
.     . . . .
.     . . . *--TRANSFER'WRITE4
.     . . .   .
.     . . .   *--CAT
.     . . .   . .
.     . . .   . *--CNVERT*
.     . . .   . .
.     . . .   . *--OUT*
.     . . .   .
.     . . .   *--IFORMAT
.     . . .   . .
.     . . .   . *--CNVERT*
.     . . .   . .
.     . . .   . *--ENCODE+
.     . . .   .
.        .  .  .     *--MAX
.        .  .  .     .
.        .  .  .     *--OUT*
.        .  .  .
.        .  .  *--SUBSTR
.        .  .     .
.        .  .     *--CAT
.        .  .     .  .
.        .  .     .  *--CNVERT*
.        .  .     .  .
.        .  .     .  *--OUT*
.        .  .     .
.        .  .     *--CNVERT*
.        .  .     .
.        .  .     *--ENCODE+.
.        .  .     .
.        .  .     *--OUT*
.        .  .
.        .  *--PH2ERR
.        .  .  .
.        .  .  *--OUT*
.        .  .  .
.        .  .  *--TRANSFER'WRITE3
.        .  .  .  .
.        .  .  .  *--CAT
.        .  .  .  .  .
.        .  .  .  .  *--CNVERT*
.        .  .  .  .  .
.        .  .  .  .  *--OUT*
.        .  .  .  .
.        .  .  .  *--IFORMAT
.        .  .  .  .  .
.        .  .  .  .  *--CNVERT*
.        .  .  .  .  .
.        .  .  .  .  *--ENCODE+
.        .  .  .  .
.        .  .  .  *--MAX
.        .  .  .  .
.        .  .  .  *--OUT*
.        .  .  .
.        .  .  *--TRANSFER'WRITE4
.        .  .     .
.        .  .     *--CAT
.        .  .     .  .
.        .  .     .  *--CNVERT*
.        .  .     .  .
.        .  .     .  *--OUT*
.        .  .     .
.        .  .     *--IFORMAT
.        .  .     .  .
.        .  .     .  *--CNVERT*
.        .  .     .  .
.        .  .     .  *--ENCODE+
.        .  .     .
.        .  .     *--MAX
.        .  .     .
.        .  .     *--OUT*
.        .  .
.        .  *--SPACES*
.        .
.        *--MAX
.        .
.        *--MIN
.        .
.        *--OUT*
.        .
.        *--PH2ERR
.        .  .
.        .  *--OUT*
.        .  .
.        .  *--TRANSFER'WRITE3
.        .  .  .
.        .  .  *--CAT
.        .  .  .  .
.        .  .  .  *--CNVERT*
.        .  .  .  .
.        .  .  .  *--OUT*
.        .  .  .
.        .  .  *--IFORMAT
.        .  .  .  .
.        .  .  .  *--CNVERT*
.        .  .  .  .
.        .  .  .  *--ENCODE+
.        .  .  .
.        .  .  *--MAX
.        .  .  .
.        .  .  *--OUT*
.        .  .
.        .  *--TRANSFER'WRITE4
.        .     .
.        .     *--CAT
.        .     .  .
.        .     .  *--CNVERT*
.        .     .  .
.        .     .  *--OUT*
.        .     .
.        .     *--IFORMAT
.        .     .  .
.        .     .  *--CNVERT*
.        .     .  .
.        .     .  *--ENCODE+
.        .     .
.        .     *--MAX
.        .     .
.        .     *--OUT*
.        .
.        *--PUSH'LAYOUT'INFO
.        .  .
.        .  *--PH2ERR
.        .     .
.        .     *--OUT*
.        .     .
.     .   *--TRANSFER'WRITE3
.     .   . .
.     .   . *--CAT
.     .   . . .
.     .   . . *--CNVERT*
.     .   . . .
.     .   . . *--OUT*
.     .   . .
.     .   . *--IFORMAT
.     .   . . .
.     .   . . *--CNVERT*
.     .   . . .
.     .   . . *--ENCODE+
.     .   . .
.     .   . *--MAX
.     .   . .
.     .   . *--OUT*
.     .   .
.     .   *--TRANSFER'WRITE4
.     .     .
.     .     *--CAT
.     .     . .
.     .     . *--CNVERT*
.     .     . .
.     .     . *--OUT*
.     .     .
.     .     *--IFORMAT
.     .     . .
.     .     . *--CNVERT*
.     .     . .
.     .     . *--ENCODE+
.     .     .
.     .     *--MAX
.     .     .
.     .     *--OUT*
.     .
.     *--REMQUO+
.
*--CONTINUE'BOX
. .
. *--ACCESS3
. . .
. . *--CAT
. . . .
. . . *--CNVERT*
. . . .
. . . *--OUT*
. . .
. . *--IFORMAT
. . . .
. . . *--CNVERT*
. . . .
. . . *--ENCODE+
. . .
. . *--MAX
. . .
.  .  '--OUT'
.  .  .
.  .  '--REMQUO+
.  .  .
.  .  '--TRANSFER'WRITE3
.  .     .
.  .     '--CAT
.  .     .  .
.  .     .  '--CNVERT'
.  .     .  .
.  .     .  '--OUT'
.  .     .
.  .     '--IFORMAT
.  .     .  .
.  .     .  '--CNVERT'
.  .     .  .
.  .     .  '--ENCODE+
.  .     .
.  .     '--MAX
.  .     .
.  .     '--OUT'
.  .
.  '--ACCESS4
.  .  .
.  .  '--CAT
.  .  .  .
.  .  .  '--CNVERT'
.  .  .  .
.  .  .  '--OUT'
.  .  .
.  .  '--IFORMAT
.  .  .  .
.  .  .  '--CNVERT'
.  .  .  .
.  .  .  '--ENCODE+
.  .  .
.  .  '--MAX
.  .  .
.  .  '--OUT'
.  .  .
.  .  '--REMQUO+
.  .  .
.  .  '--TRANSFER'WRITE4
.  .     .
.  .     '--CAT
.  .     .  .
.  .     .  '--CNVERT'
.  .     .  .
.  .     .  '--OUT'
.  .     .
.  .     '--IFORMAT
.  .     .  .
.  .     .  '--CNVERT'
.  .     .  .
.  .     .  '--ENCODE+
.  .     .
.  .     *--MAX
.  .     .
.  .     *--OUT*
.  .
.  *--CAT
.  .  .
.  .  *--CNVERT*
.  .  .
.  .  *--OUT*
.  .
.  *--IFORMAT
.  .  .
.  .  *--CNVERT*
.  .  .
.  .  *--ENCODE+
.  .
.  *--MAX
.  .
.  *--OUT*
.
*--CREATE*FILE4*RECS
.  .
.  *--ACCESS2
.  .  .
.  .  *--CAT
.  .  .  .
.  .  .  *--CNVERT*
.  .  .  .
.  .  .  *--OUT*
.  .  .
.  .  *--IFORMAT
.  .  .  .
.  .  .  *--CNVERT*
.  .  .  .
.  .  .  *--ENCODE+
.  .  .
.  .  *--OUT*
.  .  .
.  .  *--REMQUD+
.  .
.  *--ACCESS4
.  .  .
.  .  *--CAT
.  .  .  .
.  .  .  *--CNVERT*
.  .  .  .
.  .  .  *--OUT*
.  .  .
.  .  *--IFORMAT
.  .  .  .
.  .  .  *--CNVERT*
.  .  .  .
.  .  .  *--ENCODE+
.  .  .
.  .  *--MAX
.  .  .
.  .  *--OUT*
.  .  .
.  .  *--RENQUO+
.  .  .
.  .  *--TRANSFER*WRITE+
.  .     .
.  .     *--CAT
.  .     .  .
.  .     .  *--CNVERT*
.  .     .  .
.  .     .  *--OUT*
.  .     .
.  .     *--IFORMAT
.  .     .  .
.  .     .  *--CNVERT*
.  .     .  .
.  .     .  *--ENCODE+
.  .     .
.  .     *--MAX
.  .     .
.  .     *--OUT*
.  .
.  *--BOX*MAP
.  .
.  *--CAT
.  .  .
.  .  *--CNVERT*
.  .  .
.  .  *--OUT*
.  .
.  *--IFORMAT
.  .  .
.  .  *--CNVERT*
.  .  .
.  .  *--ENCODE+
.  .
.  *--MAX
.  .
.  *--OUT*
.  .
.  *--SPACES*
.  .
.  *--SUBSTR
.     .
.     *--CAT
.     .  .
.     .  *--CNVERT*
.     .  .
.     .  *--OUT*
.     .
.     *--CNVERT*
.     .
.     *--ENCODE+
.     .
.     *--OUT*
.
```

INVOCATION DIAGRAM OF THE DESIGN DIAGRAM GENERATOR (DDG)

```
*--CREATE'H'PTR'REC
.  .
.  *--ACCESS3
.  .  .
.  .  *--CAT
.  .  .  .
.  .  .  *--CNVERT*
.  .  .  .
.  .  .  *--OUT*
.  .  .
.  .  *--IFORMAT
.  .  .  .
.  .  .  *--CNVERT*
.  .  .  .
.  .  .  *--ENCODE+
.  .  .
.  .  *--MAX
.  .  .
.  .  *--OUT*
.  .  .
.  .  *--REMQJD+
.  .  .
.  .  *--TRANSFER'WRITE3
.  .     .
.  .     *--CAT
.  .     .  .
.  .     .  *--CNVERT*
.  .     .  .
.  .     .  *--OUT*
.  .     .
.  .     *--IFORMAT
.  .     .  .
.  .     .  *--CNVERT*
.  .     .  .
.  .     .  *--ENCODE+
.  .     .
.  .     *--MAX
.  .     .
.  .     *--OUT*
.  .
.  *--CAT
.  .  .
.  .  *--CNVERT*
.  .  .
.  .  *--OUT*
.  .
.  *--IFORMAT
.  .  .
.  .  *--CNVERT*
.  .  .
.  .  *--ENCODE+
.  .
.  *--OUT*
.
*--CREATE'V'PTR'REC
.  .
.  *--ACCESS3
.  .  .
.  .  *--CAT
.  .  .  .
.  .  .  *--CNVERT*
.  .  .  .
.  .  .  *--OUT*
.  .  .
.  .  *--IFORMAT
.  .  .  .
.  .  .  *--CNVERT*
.  .  .  .
.  .  .  *--ENCODE+
.  .  .
.  .  *--MAX
.  .  .
.  .  *--OUT*
.  .  .
.  .  *--REMQJO+
.  .  .
.  .  *--TRANSFER'WRITE3
.  .     .
.  .     *--CAT
.  .     .  .
.  .     .  *--CNVERT*
.  .     .  .
.  .     .  *--OUT*
.  .     .
.  .     *--IFORMAT
.  .     .  .
.  .     .  *--CNVERT*
.  .     .  .
.  .     .  *--ENCODE+
.  .     .
.  .     *--MAX
.  .     .
.  .     *--OUT*
.  .
.  *--CAT
.  .  .
.  .  *--CNVERT*
.  .  .
.  .  *--OUT*
.  .
.  *--IFORMAT
.  .  .
.  .  *--CNVERT*
.  .  .
.  .  *--ENCODE+
.  .
.  *--OUT*
.
*--GET'F1'REC
.  .
.  *--ACCESS1
.  .  .
```

```
          .  .  .--CAT
          .  .  .  .
          .  .  .  .--CNVERT'
          .  .  .  .
          .  .  .  .--OUT'
          .  .  .
          .  .  .--IFORMAT
          .  .  .  .
          .  .  .  .--CNVERT'
          .  .  .  .
          .  .  .  .--ENCODE+
          .  .  .
          .  .  .--OUT'
          .  .  .
          .  .  .--REMQUO+
          .  .
          .  .--OUT'
          .
          .--IFORMAT
          .  .
          .  .--CNVERT'
          .  .
          .  .--ENCODE+
          .
          .--INITIATE'RECORD
          .  .
          .  .--ACCESS3
          .  .  .
          .  .  .--CAT
          .  .  .  .
          .  .  .  .--CNVERT'
          .  .  .  .
          .  .  .  .--OJT'
          .  .  .
          .  .  .--IFORMAT
          .  .  .  .
          .  .  .  .--CNVERT'
          .  .  .  .
          .  .  .  .--ENCODE+
          .  .  .
          .  .  .--MAX
          .  .  .
          .  .  .--OUT'
          .  .  .
          .  .  .--REMQUO+
          .  .  .
          .  .  .--TRANSFER'WRITE3
          .  .     .
          .  .     .--CAT
          .  .     .  .
          .  .     .  .--CNVERT'
          .  .     .  .
          .  .     .  .--OUT'
          .  .     .
          .  .     .--IFORMAT
          .  .     .  .
```

```
.  .     .  .--CNVERT*
.  .     .  .
.  .     .  .--ENCODE+
.  .     .
.  .     .--MAX
.  .     .
.  .     .--OUT*
.  .
.  .--ACCESS4
.  .  .
.  .  .--CAT
.  .  .  .
.  .  .  .--CNVERT*
.  .  .  .
.  .  .  .--OUT*
.  .  .
.  .  .--IFORMAT
.  .  .  .
.  .  .  .--CNVERT*
.  .  .  .
.  .  .  .--ENCODE+
.  .  .
.  .  .--MAX
.  .  .
.  .  .--OUT*
.  .  .
.  .  .--REMQUO+
.  .  .
.  .  .--TRANSFER*WRITE4
.  .     .
.  .     .--CAT
.  .     .  .
.  .     .  .--CNVERT*
.  .     .  .
.  .     .  .--OUT*
.  .     .
.  .     .--IFORMAT
.  .     .  .
.  .     .  .--CNVERT*
.  .     .  .
.  .     .  .--ENCODE+
.  .     .
.  .     .--MAX
.  .     .
.  .     .--OUT*
.  .
.  .--BOX*MAP
.
.--INSERT
.  .
.  .--EXTRACT*NAME
.  .  .
.  .  .--ACCESS2
.  .  .  .
.  .  .  .--CAT
.  .  .  .  .
```

```
           .  .  .  .  .--CNVERT*
           .  .  .  .  .
           .  .  .  .  .--OUT*
           .  .  .  .
           .  .  .  .--IFORMAT
           .  .  .  .  .
           .  .  .  .  .--CNVERT*
           .  .  .  .  .
           .  .  .  .  .--ENCODE+
           .  .  .  .
           .  .  .  .--OUT*
           .  .  .  .
           .  .  .  .--REMQUO+
           .  .  .
           .  .  .--CAT
           .  .  .  .
           .  .  .  .--CNVERT*
           .  .  .  .
           .  .  .  .--OUT*
           .  .  .
           .  .  .--CNVERT*
           .  .  .
           .  .  .--LENGTH
           .  .  .  .
           .  .  .  .--CNVERT*
           .  .  .
           .  .  .--OUT*
           .  .  .
           .  .  .--PH2ERR
           .  .  .  .
           .  .  .  .--OUT*
           .  .  .  .
           .  .  .  .--TRANSFER'WRITE3
           .  .  .  .  .
           .  .  .  .  .--CAT
           .  .  .  .  .  .
           .  .  .  .  .  .--CNVERT*
           .  .  .  .  .  .
           .  .  .  .  .  .--OUT*
           .  .  .  .  .
           .  .  .  .  .--IFORMAT
           .  .  .  .  .  .
           .  .  .  .  .  .--CNVERT*
           .  .  .  .  .  .
           .  .  .  .  .  .--ENCODE+
           .  .  .  .  .
           .  .  .  .  .--MAX
           .  .  .  .  .
           .  .  .  .  .--OUT*
           .  .  .  .
           .  .  .  .--TRANSFER'WRITE4
           .  .  .     .
           .  .  .     .--CAT
           .  .  .     .  .
           .  .  .     .  .--CNVERT*
           .  .  .     .  .
```

```
.  .  .     .  *--OUT*
.  .  .     .
.  .  .     *--IFORMAT
.  .  .     .  .
.  .  .     .  *--CNVERT*
.  .  .     .  .
.  .  .     .  *--ENCODE*
.  .  .     .
.  .  .     *--MAX
.  .  .     .
.  .  .     *--OUT*
.  .  .
.  .  *--SUBSTR
.  .     .
.  .     *--CAT
.  .     .  .
.  .     .  *--CNVERT*
.  .     .  .
.  .     .  *--OUT*
.  .     .
.  .     *--CNVERT*
.  .     .
.  .     *--ENCODE*
.  .     .
.  .     *--OUT*
.  .
.  *--PM2ERR
.  .  .
.  .  *--OUT*
.  .  .
.  .  *--TRANSFER*WRITE3
.  .  .  .
.  .  .  *--CAT
.  .  .  .  .
.  .  .  .  *--CNVERT*
.  .  .  .  .
.  .  .  .  *--OUT*
.  .  .  .
.  .  .  *--IFORMAT
.  .  .  .  .
.  .  .  .  *--CNVERT*
.  .  .  .  .
.  .  .  .  *--ENCODE*
.  .  .  .
.  .  .  *--MAX
.  .  .  .
.  .  .  *--OUT*
.  .  .
.  .  *--TRANSFER*WRITE4
.  .     .
.  .     *--CAT
.  .     .  .
.  .     .  *--CNVERT*
.  .     .  .
.  .     .  *--OUT*
```

```
*   *       *--IFORMAT
*   *       *   *
*   *       *   *--CNVERT*
*   *       *   *
*   *       *   *--ENCODE+
*   *       *
*   *       *--MAX
*   *       *
*   *       *--OUT*
*   *
*   *--SPACES*
*
*--OUT*
*
*--POP*LAYOUT*INFO
*   *
*   *--PH2ERR
*       *
*       *--OUT*
*       *
*       *--TRANSFER*WRITE3
*       *   *
*       *   *--CAT
*       *   *   *
*       *   *   *--CNVERT*
*       *   *   *
*       *   *   *--OUT*
*       *   *
*       *   *--IFORMAT
*       *   *   *
*       *   *   *--CNVERT*
*       *   *   *
*       *   *   *--ENCODE+
*       *   *
*       *   *--MAX
*       *   *
*       *   *--OUT*
*       *
*       *--TRANSFER*WRITE4
*           *
*           *--CAT
*           *   *
*           *   *--CNVERT*
*           *   *
*           *   *--OUT*
*           *
*           *--IFORMAT
*           *   *
*           *   *--CNVERT*
*           *   *
*           *   *--ENCODE+
*           *
*           *--MAX
*           *
*           *--OUT*
*
```

```
'--POP'PROC'STACK
.  .
.  '--CAT
.  .  .
.  .  '--CNVERT*
.  .  .
.  .  '--OUT*
.  .
.  '--IFORMAT
.  .  .
.  .  '--CNVERT*
.  .  .
.  .  '--ENCODE+
.  .
.  '--OUT*
.  .
.  '--PH2ERR
.     .
.     '--OUT*
.     .
.     '--TRANSFER'WRITE3
.     .  .
.     .  '--CAT
.     .  .  .
.     .  .  '--CNVERT*
.     .  .  .
.     .  .  '--OUT*
.     .  .
.     .  '--IFORMAT
.     .  .  .
.     .  .  '--CNVERT*
.     .  .  .
.     .  .  '--ENCODE+
.     .  .
.     .  '--MAX
.     .  .
.     .  '--OJT*
.     .
.     '--TRANSFER'WRITE4
.        .
.        '--CAT
.        .  .
.        .  '--CNVERT*
.        .  .
.        .  '--OUT*
.        .
.        '--IFORMAT
.        .  .
.        .  '--CNVERT*
.        .  .
.        .  '--ENCODE+
.        .
.        '--MAX
.        .
.        '--OUT*
.
```

```
C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER
INVOCATION DIAGRAM OF THE DESIGN DIAGRAM GENERATOR (DDG)

'--PUSH'LAYOUT'INFO
. .
. '--PH2ERR
.   .
.   '--OUT'
.   .
.   '--TRANSFER'WRITE3
.   . .
.   . '--CAT
.   . . .
.   . . '--CNVERT'
.   . . .
.   . . '--OUT'
.   . .
.   . '--IFORMAT
.   . . .
.   . . '--CNVERT'
.   . . .
.   . . '--ENCODE+
.   . .
.   . '--MAX
.   . .
.   . '--OUT'
.   .
.   '--TRANSFER'WRITE4
.     .
.     '--CAT
.     . .
.     . '--CNVERT'
.     . .
.     . '--OUT'
.     .
.     '--IFORMAT
.     . .
.     . '--CNVERT'
.     . .
.     . '--ENCODE+
.     .
.     '--MAX
.     .
.     '--OUT'
.
'--PUSH'PROC'STACK
  .
  '--CAT
  . .
  . '--CNVERT'
  . .
  . '--OUT'
  .
  '--IFORMAT
  . .
  . '--CNVERT'
  . .
  . '--ENCODE+
  .
```

```
          '--OUT'
          .
          '--PH2ERR
             .
             '--OUT'
             .
             '--TRANSFER'WRITE3
             .  .
             .  '--CAT
             .  .  .
             .  .  '--CNVERT'
             .  .  .
             .  .  '--OUT'
             .  .
             .  '--IFORMAT
             .  .  .
             .  .  '--CNVERT'
             .  .  .
             .  .  '--ENCODE+
             .  .
             .  '--MAX
             .  .
             .  '--OUT'
             .
             '--TRANSFER'WRITE4
                .
                '--CAT
                .  .
                .  '--CNVERT'
                .  .
                .  '--OUT'
                .
                '--IFORMAT
                .  .
                .  '--CNVERT'
                .  .
                .  '--ENCODE+
                .
                '--MAX
                .
                '--OUT'

CONTINUATIONS AND INDEPENDENT ROUTINES

 --NULL
   .
   '--CNVERT'
```

MISSION
of
Rome Air Development Center

RADC plans and conducts research, exploratory and advanced development programs in command, control, and communications (C^3) activities, and in the C^3 areas of information sciences and intelligence. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.

AMERICAN REVOLUTION BICENTENNIAL 1776-1976